

Symbolic Analysis of Large Analog Integrated Circuits: The Numerical Reference Generation Problem

*Francisco V. Fernández, Oscar Guerra, Juan D. Rodríguez-García and Angel
Rodríguez-Vázquez*

Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica
E-41012-Sevilla, SPAIN

Footnote

This work has been supported by the EEC ESPRIT Program in the Framework of the Project #21812 (AMADEUS) and the Spanish C.I.C.Y.T. under contract TIC97-0580.

Abstract

Symbolic analysis potentialities for gaining circuit insight and for efficient repetitive evaluations have been limited by the exponential increase of formula complexity with the circuit size. This drawback has began to be solved by the introduction of simplification before and during generation techniques. An appropriate error control in both involves the generation of a numerical reference, which implies the calculation of network functions in the complex frequency variable. The polynomial interpolation method, traditionally used for this task, is analyzed in detail, its limitations for large circuit analysis are pointed out, and an adaptive scaling mechanism is proposed to meet the efficiency and accuracy requirements imposed by the new simplification methodologies.

I. INTRODUCTION

Symbolic circuit analysis refers to the calculation of network functions where the complex frequency and all or part of the circuit parameters are symbols. These functions are typically given in the form:

$$H(s, \mathbf{x}) = \frac{\sum_{i=0}^N s^i f_i(\mathbf{x})}{\sum_{j=0}^M s^j g_j(\mathbf{x})} \quad (1)$$

where $f_i(\mathbf{x})$ and $g_j(\mathbf{x})$ are sums of products of the symbolic parameters $\mathbf{x}^T = \{x_1, x_2, \dots, x_Q\}$. See for instance [1] and [2] for an actualized review of techniques and applications of symbolic analysis.

Plain symbolic analysis suffers from a tremendous increase of expression complexity with the circuit size. Consider for illustration's sake the circuits in Fig. 1. The DC voltage gain of Fig. 1a using the model in Fig. 1b is:

$$\frac{v_o}{v_i} = [g_{m1}g_{m2}r_{\pi1}r_{\pi2}R_3R_L(R_1 + R_2) + R_1R_L(R_3 + r_{\pi2})g_{m1}r_{\pi1} + R_1R_L(R_3 + r_{\pi2})] / [g_{m1}g_{m2}r_{\pi1}r_{\pi2}R_3R_LR_1 + g_{m1}r_{\pi1}(R_2 + R_L)(R_3 + r_{\pi2})R_1 + (R_2 + R_L)(R_3 + r_{\pi2})(R_1 + r_{\pi1}) + (R_3 + r_{\pi2})r_{\pi1}R_1] \quad (2)$$

which contains 21 terms; this number raises to 8616 for the Miller opamp in Fig. 1c using the model in Fig. 1d, and is well above 10^{16} for the $\mu A741$ opamp in Fig. 1e using the model in Fig. 1f¹. Given this exponential increase of the term count with the number of elements in the circuit model, symbolic expression *simplification* has been recognized to be essential for both: *formula interpretation* by human designers and computer manipulation for *repetitive evaluations* in design automation applications [1]. For instance, elimination of the least significant terms in (2) leads to $\frac{v_o}{v_i} \approx \frac{(R_1 + R_2)}{R_1}$ which is a much more interpretable expression.

Conventional simplification approaches first calculate the *complete* symbolic expression, and then simplify it by eliminating insignificant terms or sub-expressions, based on numerical estimates of the symbolic parameters—commonly called Simplification After Generation (SAG). Consequently, most of the resources employed to generate the pruned terms are wasted. Besides, although this is a feasible approach for circuits like those of Fig. 1a and c (in general,

1. The number of terms for Fig. 1c was obtained using ASAP [3] while the lower bound of the number of terms for Fig. 1e was calculated using the theory presented in [4].

for circuits with less than around 50 symbols), it is unfeasible for circuits like that in Fig. 1e, as no computer has enough memory to handle such a huge number of symbolic terms, on the one hand, and the time needed to generate them would not be affordable, on the other. These larger circuits have to be analyzed by using the newest approaches: Simplification During Generation (SDG) and Simplification Before Generation (SBG). This paper deals with a basic ingredient of these new techniques, namely the generation of a numerical reference to evaluate the errors in the simplification process. Based on a brief description of our implementation of SBG and SDG (Section II), Section III addresses the generation of this reference, describing the problems arising when handling medium and large size analog integrated circuits and introducing new algorithms for its efficient calculation. Experimental results are shown in Section IV.

II. THE APPROXIMATION METHODOLOGY

A. Simplification During Generation

SDG techniques start from some formulation of the network equations and solve them trying to directly generate the simplified expression. In our SDG approach, symbolic terms are generated in decreasing order of magnitude until the generated terms represent a significant fraction of the complete expression.

The first reliable algorithms capable of efficiently generating terms in decreasing order of magnitude [5]-[8] were based on the *two-graph* method [9]. The computation of the simplified coefficients of s^k reduces to the following problem: “Given the voltage graph G_V and the current graph G_I of a circuit with n nodes, enumerate subsets of $(n - 1)$ branches in decreasing order of magnitude such that: (a) form a spanning tree in G_V ; (b) form a spanning tree in G_I ; and, (c) contain k capacitances and $(n - k - 1)$ (trans)conductances”. This problem can be formulated in terms of *matroids* [10]. Each condition (a)–(c) above is mapped into a matroid and the problem of generation of common spanning trees in order is mapped into a weighted matroid intersection problem. The algorithms in [5]-[8] calculate the intersection of two matroids among (a)-(c) [11], and then check if it intersects the third matroid. Although the intersection problem of three general matroids is nonpolynomial-hard, [12], [13] have reported the first algorithm able to solve it by exploiting the characteristics of the three particular matroids at hand. These algorithms have made feasible the analysis of large circuits like the $\mu A741$ opamp in a few tens of seconds.

An important ingredient of SDG is the error criterion used to stop the generation of terms.

Consider that $h_k(\mathbf{x}) = \sum_{l=1}^T h_{kl}(\mathbf{x})$ represents either $f_i(\mathbf{x})$ or $g_j(\mathbf{x})$ in (1). The P most significant

terms are generated in $h_k(\mathbf{x})$ until the sum of the generated terms represents a given fraction of the total magnitude of the coefficient,

$$\left| h_k(\mathbf{x}_o) - \sum_{l=1}^P h_{kl}(\mathbf{x}_o) \right| < \varepsilon_k |h_k(\mathbf{x}_o)| \quad (3)$$

where \mathbf{x}_o represents a design point of the circuit parameters and ε_k is an error control parameter which is obtained by backpropagation from maximum magnitude and phase error specifications. As shown in (3), the total magnitude of each circuit coefficient, $h_k(\mathbf{x}_o)$, must be known a priori; however, the fully symbolic expression is not available for such calculation. Hence, an efficient technique able to calculate (1) with only s as symbolic variable is needed. The problems arising in this calculation when handling large circuits are addressed in Section III.

Once an approximated expression $H_{ap}(s)$ has been calculated, the maximum magnitude and phase errors with respect to the exact expression $H_{ex}(s)$ in a given frequency range can be obtained from:

$$\varepsilon_{|H|} = \frac{|H_{ex}(j\omega)| - |H_{ap}(j\omega)|}{|H_{ex}(j\omega)|} = 1 - \frac{\sqrt{\frac{N_{apr}^2 + N_{api}^2}{D_{apr}^2 + D_{api}^2}}}{\sqrt{\frac{N_{exr}^2 + N_{exi}^2}{D_{exr}^2 + D_{exi}^2}}} \quad (4)$$

$$\Delta\phi_H = \angle H_{ex}(j\omega) - \angle H_{ap}(j\omega) = \text{atan} \frac{N_{exi}}{N_{exr}} - \text{atan} \frac{D_{exi}}{D_{exr}} - \text{atan} \frac{N_{api}}{N_{apr}} + \text{atan} \frac{D_{api}}{D_{apr}}$$

where

$$\begin{aligned} H_{ap}(j\omega) &= \frac{N_{ap}(j\omega)}{D_{ap}(j\omega)} = \frac{N_{apr}(\omega) + jN_{api}(\omega)}{D_{apr}(\omega) + jD_{api}(\omega)} \\ H_{ex}(j\omega) &= \frac{N_{ex}(j\omega)}{D_{ex}(j\omega)} = \frac{N_{exr}(\omega) + jN_{exi}(\omega)}{D_{exr}(\omega) + jD_{exi}(\omega)} \end{aligned} \quad (5)$$

The application of interval analysis techniques [14] to (4) to evaluate the maximum magnitude and phase errors in a given frequency range usually yields overly conservative estimates of those maxima. Therefore, interval analysis techniques are applied to the derivatives of (4) to delimit frequency subranges in which the maximum magnitude and phase errors occur. Then, the frequency points for which the maximum magnitude or phase error occurs in those frequency subranges are easily calculated using the Newton-Raphson method.

B. Simplification Before Generation

SBG performs the approximation during the set-up of the network equations by eliminating matrix entries, graph branches, etc. Then, the reduced matrix or graph is much easier to solve. Our SBG approach takes place at the graph level, replacing those elements, whose contribution (appropriately measured) to the network function is negligible, with a zero-admittance or zero-impedance element. This is illustrated with the simple example in Fig. 2. Assuming that G_2 is evaluated to have the smallest contribution to the network function i_o/v_i , it can be deleted from the voltage and current graphs. The network function for the simplified circuit is

$$\frac{i_o}{v_i} = \frac{C_3 s}{1 + (R_1 C_3 + R_4 C_3) s} \quad (6)$$

which is significantly less complex than that resulting for the original graphs:

$$\frac{i_o}{v_i} = \frac{R_2 C_3 s}{(R_1 + R_2) + (R_1 R_2 C_3 + R_1 R_4 C_3 + R_2 R_4 C_3) s} \quad (7)$$

The same deletion/contraction operation is repeated for the next element with smallest contribution, and so on. The reduction in formula complexity is more significant for larger circuits.

Reported approaches evaluate the influence of the elimination of matrix entries [15], [16] or graph branches [7] at a single or at a finite number of sample frequency points, and hence do not guarantee accuracy at other frequency points. To solve this problem we evaluate each element contribution by comparing the network function of the complete circuit and that of a modified circuit in which the element has been deleted/contracted. This implies calculating the network function as a function of s for each deletion/contraction. The polynomial interpolation method, which is considered to be the most efficient one to perform this task, is analyzed in detail in Section III. Detection of the maximum magnitude and phase errors induced by a device replacement is performed as described in Section IIA.

However, even the techniques presented in Section III to improve the efficiency of this method are not sufficient for a repetitive application in SBG. But, it usually turns out that many numerator and denominator coefficients do not have a significant contribution in the frequency range of interest. A large error in those coefficients is unimportant and, hence, they can be neglected. For instance, once the coefficients of the voltage gain of the $\mu A741$ opamp have been calculated, all those which only become significant above 10MHz can be neglected as the opamp will never be operated at such frequency. This operation drastically reduces the cost of

subsequent network function calculations as the polynomial interpolation cost grows with the number of network function coefficients.

The flow diagram of the complete symbolic analysis methodology including SBG and SDG is shown in Fig. 3. It must be noticed that the SBG step makes the SDG task much more effective but it is not essential, that is, SDG can do the work without a previous SBG step.

III. NETWORK FUNCTIONS IN s

Section II has shown that error evaluation in both, SBG and SDG techniques, requires repetitive calculation of network functions with s as the unique symbolic variable for complete or reduced circuit models. This establishes how important is to develop efficient calculation techniques of network functions of medium and large size circuits. The polynomial interpolation method is considered to be one of the most efficient techniques to perform this task [9],[17].

A. Background on polynomial interpolation

The polynomial interpolation starts from the fact that the coefficients of a n -th order polynomial,

$$P(s) = p_0 + p_1s + p_2s^2 + \dots + p_ns^n \quad (8)$$

can be obtained from the polynomial values at $(n+1)$ distinct points $P(s_k)$. If these values can be calculated, then the following matrix equation can be formulated:

$$\begin{bmatrix} 1 & s_0 & s_0^2 & \dots & s_0^n \\ 1 & s_1 & s_1^2 & \dots & s_1^n \\ \dots & & & & \\ 1 & s_n & s_n^2 & \dots & s_n^n \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_n \end{bmatrix} = \begin{bmatrix} P(s_0) \\ P(s_1) \\ \dots \\ P(s_n) \end{bmatrix} \quad (9)$$

The matrix in (9) is nonsingular and hence (9) has always a unique solution. Such solution is the set of polynomial coefficients, p_i , in (8).

It has been shown that the use of $K \geq n + 1$ equally-spaced interpolation points in the unit circle gives the best results concerning numerical accuracy and stability [9], [17]. Once the values of (8) at all these points $P(s_k)$ are known, the polynomial coefficients can be obtained through the *Discrete Fourier Transform* (DFT),

$$\hat{p}_i = \frac{1}{K} \sum_{k=0}^{K-1} P(s_k) e^{-\frac{2\pi i k}{K}} \quad i = 0, 1, \dots, K-1 \quad (10)$$

where

$$\hat{p}_i = \begin{cases} p_i & \text{for } i \leq n \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The number of interpolation points, K , should be at least $(n+1)$, but in most cases, like that we are dealing with, the polynomial order n is not known beforehand. Hence, an upper estimate on K must be done, and (10) should be identically 0 for those coefficients over the n -th power.

Our objective is not the calculation of a polynomial but a network function, which is given by the ratio of two polynomials. Therefore, the polynomial interpolation method is applicable to our problem once the values of the numerator $N(s_k)$ and denominator $D(s_k)$ at the different interpolation points are known. In order to calculate $N(s_k)$ and $D(s_k)$ assume that an appropriate formulation method, i.e. modified nodal analysis, has been applied on the circuit so that the network equations can be written as:

$$\mathbf{Y}_{MNA} \mathbf{X} = \mathbf{E} \quad (12)$$

where \mathbf{Y}_{MNA} is the modified nodal matrix, \mathbf{X} contains nodal voltages and auxiliary currents, and \mathbf{E} accounts for the influence of the independent sources. Once any frequency-dependent element in \mathbf{Y}_{MNA} is evaluated at the interpolation point $s=s_k$, the value of the network function

$$H(s_k) = \frac{N(s_k)}{D(s_k)} \quad (13)$$

can be obtained by applying LU decomposition and backsubstitution to (12). The denominator of the network function is easily obtained as:

$$D(s_k) = |\mathbf{Y}_{MNA}(s_k)| \quad (14)$$

and the numerator $N(s_k)$ is easily obtained from (13) and (14):

$$N(s_k) = H(s_k) \cdot D(s_k) \quad (15)$$

B. Introducing scaling

One major problem in polynomial interpolation applied to analog integrated circuits is the

dramatic effect of round-off errors, due to the finite precision arithmetics of computers. The calculation of the differential voltage gain as a function of s in the positive feedback OTA of Fig. 4a with the transistor model in Fig. 4b shows these problems. The order of this network function is unknown a priori, but an upper bound of the order can be estimated at 9; hence 10 interpolation points are used. The interpolated numerator and denominator coefficients when using interpolation points located at the unit circle are given in Table 1.

Polynomial coefficients must be real, but, as shown in Table 1, many interpolated coefficients have a non-zero imaginary component. This is due to the round-off errors, which avoid perfect cancellations of the imaginary parts in the DFT. The values of these imaginary components give us an idea of the numerical noise level induced by the finite number of bits available in a digital computer to represent the floating point numbers. The real and imaginary parts of the interpolated coefficients in Table 1, except the two shadowed ones, are of the same order of magnitude; therefore, the actual value of those coefficients is not obtained as it is below the numerical error level. Also, as indicated by (11) the zero coefficients would indicate the actual polynomial order, but as shown in Table 1 the zero coefficients are not detected now.

The numerical error level in the polynomial interpolation depends on the coefficient p_i in (8) having the largest absolute value. This error level is about $10^{-13} \times \max_i |p_i|$ in a computer with 16-decimal-digit accuracy [17],[18]. The spread of values between the maximum and minimum coefficient should be well below this error to ensure numerical accuracy of the calculated coefficients:

$$\frac{\min_i |\text{coefficient}_i|}{\max_i |\text{coefficient}_i|} \gg 10^{-13} \quad (16)$$

Hence, it is not difficult to see that the second and higher order coefficients in Table 1 are not valid.

Each polynomial coefficient in typical analog integrated circuits is a sum-of-products of admittances: (trans)conductances and capacitances. Therefore, the coefficient of s^i has one more (trans)conductance and one less capacitance in each term than the coefficient of s^{i+1} . Taking into account the typical magnitudes of (trans)conductances and capacitances in analog circuits we can expect an extremely large spread of coefficient values. In order to reduce it, the complex frequency variable (equivalently the capacitor values) should be scaled before performing the polynomial interpolation on the unit circle [18]. Also, this suggests conductance scaling as another alternative.

The effect of conductance and frequency scaling on the relative value of the coefficients is analogous. Assume that g is the conductance scale factor and f is the frequency scale factor, then the polynomial with normalized coefficients p'_i is:

$$P(s) = \sum_{i=0}^n p'_i s^i = \sum_{i=0}^n p_i f^i g^{L-i-1} s^i = g^{L-1} \sum_{i=0}^n p_i \left(\frac{f}{g}\right)^i s^i \quad (17)$$

where L is the number of AC nodes in the circuit. We can observe that increasing the frequency scale factor has the same effect as decreasing the conductance scale factor.

For illustration's sake, Table 2a shows the normalized denominator coefficients, which were obtained using a frequency scale factor $f = 10^8$. The shadowed coefficients are well above the error level $10^{-13} \times \max_i |p_i|$, where the maximum coefficient is the second-order one, dark-shadowed in Table 2. Therefore, all shadowed coefficients can be considered to be correct. Table 2b shows the denominator coefficients after frequency denormalization. An equivalent result is obtained for the numerator.

C. Adaptive scaling

A major problem is that the appropriate scale factor is not known a priori. Moreover, for larger circuits (and correspondingly higher order network functions) no scale factor can be found that keeps the spread of coefficient values under the error level.

The solution to this problem must clearly go through the use of several scale factors. [18] proposed the selection of several frequency scale factors to obtain sets of valid coefficients. However, Section II has shown that SBG and SDG methodologies require extremely efficient techniques for the numerical reference generation. Therefore, the tentative proposal and test of many scale factors is not a viable solution.

Our objective is to make a minimum number of interpolations. The proposed solution is an iterative mechanism which performs successive interpolations using an adaptive scaling mechanism as schematically shown in the flow diagram in Fig. 5. The algorithm operation at the i -th operation is illustrated in Fig. 6.

After each polynomial interpolation a region of valid coefficients is selected. Assume that p_k 's are the interpolated coefficients (normalized by the scale factors g_i and f_i) at iteration i and p_m is the polynomial coefficient with the largest absolute value (dark-shadowed in Fig. 6). If coefficients with σ significant digits need to be calculated, then all coefficients p_k which prior to denormalization are

$$|p_k| < 10^{-13+\sigma} \times \max_i |p_i| = 10^{-13+\sigma} \times |p_m| \quad (18)$$

must be neglected. Assume that p_b and p_e limit the region of coefficients which meet the condition in (18). All coefficients in that region are considered to be valid and marked as calculated (light-shadowed in Fig. 6). Based on these results two new scale factors f_{i+1} and g_{i+1} are calculated for the following polynomial interpolation as follows:

$$g_{i+1} = \frac{g_i}{\sqrt{q}} \quad f_{i+1} = f_i \sqrt{q} \quad (19)$$

where

$$|p_b|q^b = |p_m|q^m \times 10^{13+r} \quad (20)$$

to shift the region of valid coefficients to smaller powers of s , being p_b and p_m the first and maximum coefficients within the last valid region respectively, and r a tuning factor; and

$$|p_e|q^e = |p_m|q^m \times 10^{13+r} \quad (21)$$

to shift the region of valid coefficients to higher powers of s , being p_e the last coefficient of the previous valid region.

The objective of the tuning factor r in (20)-(21) is that the region of valid coefficients at the $(i+1)$ -th iteration has the smallest overlapping with the region at the i -th iteration. To clarify the effect of r , consider for instance the use of (19) and (21) to shift the region of valid coefficients to higher powers of s . $|p_e|$ is slightly larger than $10^{-13+\sigma} \times |p_m|$. By choosing $r = \sigma$ the scale factor q given by (21) makes that the lower value for the region of valid coefficients in the subsequent iteration is approximately $|p_e|$. Since this is an approximation and to avoid that one coefficient remains uncalculated between the two consecutive valid regions the tuning factor r is heuristically chosen smaller: $r = \sigma - 1$. A more conservative selection, $r \ll \sigma$, is not advisable because that would make many coefficients to be redundantly calculated in subsequent interpolations with the corresponding extra computational effort

The scale factors for the first interpolation are calculated as a function of the mean value of capacitances and (trans)conductances. The objective of these heuristics is to first generate the widest region of valid coefficients.

If between two consecutive valid regions, with scale factors f_j, g_j and f_l, g_l , some incorrect coefficients remain, then new scale factors f_{i+1}, g_{i+1} , are calculated as follows,

$$f_{i+1} = f_j \cdot f_l \qquad g_{i+1} = g_j \cdot g_l \qquad (22)$$

Notice that in the algorithm above simultaneous scaling of both, frequency and conductance is used. This technique is used to avoid using too large ($> \sim 10^{18}$) frequency or conductance scale factors. These high values occasionally occur when using a single scale factor and are responsible for an increase of the error in the calculation of numerator and denominator of the transfer function at the interpolation points.

D. Problem reduction mechanism

In each polynomial interpolation the computational effort depends on the number of interpolation frequencies needed. The problem complexity can be reduced at subsequent iterations of previous algorithm, once the coefficients of the highest or smallest powers of s have been calculated. Assume the coefficients $p_0 \dots p_{k-1}$ and $p_{l+1} \dots p_n$ have already been calculated, then the polynomial is transformed as follows,

$$P'(s) = p_k + \dots + p_l \cdot s^{l-k} = \frac{P(s) - \sum_{i=0}^{k-1} p_i \cdot s^i - \sum_{i=l+1}^n p_i \cdot s^i}{s^k} \qquad (23)$$

The new polynomial contains the coefficients that still have to be calculated and needs only $l-k+1$ interpolation points. This simple operation drastically reduces the computation time at subsequent iterations.

IV. EXPERIMENTAL RESULTS

The proposed algorithm is applied in this section to two examples: the analysis of the $\mu A741$ opamp and a bandpass biquad described at the transistor level.

A. The $\mu A741$ opamp

Consider the voltage gain of the $\mu A741$ opamp in Fig. 1e with the small-signal BJT model of Fig. 1f. The results of the polynomial interpolation with the first frequency and conductance scale factors are partially shown in Table 3. The imaginary parts of the coefficients have been omitted from Table 3 as they originate from the numerical noise in the DFT. According to (18), the region of valid denominator coefficients is determined from the coefficient with largest normalized absolute value (dark-shadowed in Table 3): p_3 . If 6 significant digits are desired all denominator coefficients larger than

$$10^{-13+6} \times 1.28095 \times 10^{124} = 1.28095 \times 10^{117} \quad (24)$$

are considered correct. That means that the region of valid denominator coefficients extends from p_0 to p_{12} (light-shadowed). The remaining coefficients are not shown in Table 3 as it lacks interest. As indicated by (13)-(15) numerator and denominator coefficients are obtained simultaneously with a minimum extra cost. Hence, a region of valid numerator coefficients is also determined analogously.

The results of the first interpolation are used to calculate new scale factors f_2 and g_2 able to provide a region of valid coefficients of higher powers of s . For this, (19) and (21) are applied using $p_e = p_{12}$ and $p_m = p_3$. The problem reduction given by (23) allows the use of 13 less interpolation points in the next polynomial interpolation. The generated coefficients are shown in Table 4. The maximum absolute value coefficient, p_{22} , and the application of (18) delimits again the region of valid coefficients, which, as shown in Table 4, has shifted to the region between the 12-th and the 34-th coefficient of the denominator. It can be seen that the overlap between the valid region in Table 3 and Table 4 reduces to one coefficient in the denominator and there is no overlap in the numerator.

Again, (19) and (21) are applied to the coefficients in Table 4 to get a new set of scale factors for higher order coefficients. The problem reduction mechanism reduces in 22 less interpolation points for the third (last) iteration of the algorithm. The remaining coefficients, obtained in the third polynomial interpolation, are shown in Table 5. In this case there is an overlap of two coefficients in numerator and denominator.

The CPU time to get the results in this example was 3.9s for the first iteration, 2.3s for the second one and 0.9s for the third one (measured on a SPARC Station 10). The decrease in the number of interpolation points due to the problem reduction mechanism is clearly reflected in a CPU time reduction at subsequent iterations.

The accuracy of the results obtained in this example is demonstrated through the comparison of the Bode diagrams obtained from the interpolation of numerator and denominator of the voltage gain of $\mu A741$ and those obtained through a commercial electrical simulator, which are shown in Fig. 7. A perfect matching appears in all the frequency range.

B. Bandpass biquad

As a second example consider the bandpass biquad in Fig. 8a with the opamps described at the transistor level, as shown in Fig. 8b. The small-signal model used for the bipolar transistors was the same as for the previous example, shown in Fig. 1f. For limited space reasons and

without loss of generality we will limit ourselves to the calculation of the denominator of the voltage gain of the biquad.

The polynomial interpolation with the first set of frequency and conductance scale factors provides a region of valid coefficients which extends from the 25-th to the 59-th coefficient, as shown in Table 6a. To shift the region of valid coefficients to smaller powers of the frequency, (19) and (20) are applied to the results of the first interpolation. No problem reduction can be performed at this iteration. However, for the scale factors used the coefficients above the 59-th one are smaller than the error level. With the new scale factors shifting the valid region to smaller powers of s , the influence of the coefficients above the 59-th one on the polynomial value will be still smaller, and, hence, can be neglected. Neglecting these high order coefficients is useful because it allows to handle the polynomial as of smaller order, reducing in this way the number of points needed in the interpolation.

The polynomial interpolation with the new set of scale factors gives a region of valid coefficients which extends from the second to the 24-th coefficient and is shown in Table 6b. The calculation of the first coefficient, which is under the numerical error level, does not need an additional polynomial interpolation but a single LU decomposition with no frequency-dependent element in the circuit.

Then, the first 60 denominator coefficients are available, the problem is reduced, and, hence, 60 less interpolation points are needed at the following iteration. Now, the region of valid coefficients must be shifted to higher powers of s ; so, (19) and (21) are applied to the results of the first iteration of the algorithm. A new polynomial interpolation gives the results shown in Table 6c, where the region of valid coefficients is light-shadowed in Table 6c and its limits have been determined by (18). Again, the results of this interpolation are used to calculate new scale factors to shift the region of valid coefficients to higher powers of s , and to reduce the number of interpolation points at the following iteration. A new polynomial interpolation gives finally the remaining denominator coefficients, shown in Table 6d.

The CPU time spent to get the results shown in Table 6 is 30s. This time rises to 80s in case the problem reduction mechanism is not used. It could be argued that the CPU time obtained in these examples is acceptable for SDG where the network function in s must be calculated only once, while it is still too high for SBG where the polynomial interpolation or network function calculation might need to be calculated hundredths of times. This is not commonly true as for real circuits the results of the first network function calculation can be used to neglect a large number of coefficients for the frequency range in which we are inter-

ested. That means that the number of interpolation points and, hence, the CPU time, is drastically reduced for the following executions of the algorithm.

V. CONCLUSIONS

This paper has addressed the problems arising in the calculation of the numerical references, essential for an accurate error control in the proposed Simplification Before and During Generation approach for symbolic analysis of large analog circuits. The proposed algorithm is based on the polynomial interpolation method and incorporates simultaneous frequency and conductance scaling, an adaptive updating of the scale factors, and a problem reduction mechanism to speed up the generation of the numerical references. The experimental results obtained with large real-life circuits demonstrate the practical applicability of the techniques introduced in the paper.

VI. REFERENCES

- [1] F.V. Fernández, A. Rodríguez-Vázquez, J.L. Huertas and G. Gielen, eds., *Symbolic Analysis Techniques and Applications to Analog Design Automation*. Piscataway, NJ: IEEE Press, 1998.
- [2] G. Gielen, P. Wambacq and W. Sansen, "Symbolic analysis methods and applications for analog circuits: A tutorial overview," *Proc. of the IEEE*, vol. 82, no. 2, pp. 287-304, February 1994.
- [3] F. V. Fernández, A. Rodríguez-Vázquez, J. D. Martín, and J. L. Huertas, "Formula approximation for flat and hierarchical symbolic analysis," *Analog Integrated Circuits and Signal Processing*, vol. 3, no. 1, pp. 43-58, January 1993.
- [4] M. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms*. New York: John Wiley and Sons, 1981.
- [5] F.V. Fernández, P. Wambacq, G. Gielen, A. Rodríguez-Vázquez and W. Sansen, "Symbolic analysis of large analog integrated circuits by approximation during expression generation," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. CAD, pp. 25-28, 1994.
- [6] P. Wambacq, F.V. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez, "Efficient symbolic computation of approximated small-signal characteristics of analog integrated circuits," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 327-330, March 1995.
- [7] Q. Yu and C. Sechen, "Approximate symbolic analysis of large analog integrated circuits," *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 664-671, 1994.
- [8] Q. Yu and C. Sechen, "Efficient approximation of symbolic network functions using matroid intersection algorithms," *Proc. IEEE Int. Symp. Circuits Systems*, pp. 2088-2091, 1995.
- [9] P.M. Lin, *Symbolic Network Analysis*. Amsterdam: Elsevier, 1991.
- [10] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.
- [11] P.M. Camerini and H.W. Hamacher, "Intersection of two matroids: (condensed) border graph and ranking", *SIAM J. Discrete Mathematics*, vol. 2, pp. 16-27, February 1989.

- [12] M. Galán, I. García-Vargas, F.V. Fernández and A. Rodríguez-Vázquez, "A new matroid intersection algorithm for symbolic large circuit analysis," *Proc. Workshop on Symbolic Methods and Applications to Circuit Design*, Leuven, Belgium, 1996.
- [13] M. Galán, F.V. Fernández and A. Rodríguez-Vázquez, "Comparison of matroid intersection algorithms for large circuit analysis," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1784-1787, 1997.
- [14] R. E. Moore, *Methods and Applications of Interval Analysis*. Studies in Applied Mathematics, Philadelphia, 1979.
- [15] Jer-Jaw Hsu and C. Sechen, "Fully symbolic analysis of large analog integrated circuits," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 21.4.1–21.4.4, 1994.
- [16] R. Sommer, E. Hennig, G. Droge, and E.-H. Horneber, "Equation-based symbolic approximation by matrix reduction with quantitative error prediction," *Alta Frequenza*, vol. 5, no. 6, pp. 317–325, November 1993.
- [17] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, 1994.
- [18] K. Singhal and J. Vlach, "Generation of immittance functions in symbolic form for lumped distributed active networks," *IEEE Trans. Circuits and Systems*, vol. CAS-21, no. 1, pp. 57-67, January 1974.

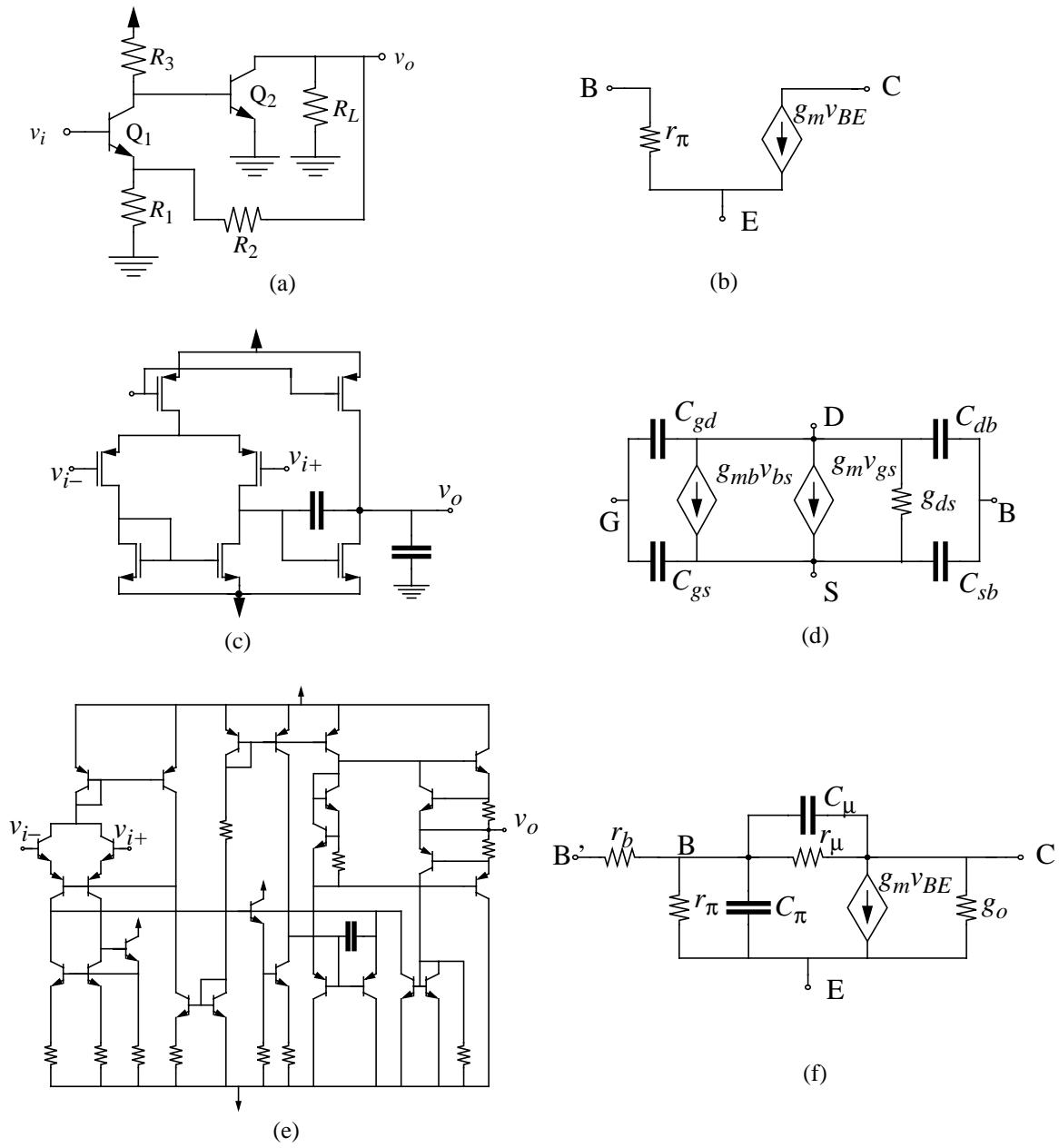


Figure 1 (a) BJT feedback amplifier; (b) low-frequency BJT model; (c) Miller operational amplifier; (d) MOSFET model; (e) $\mu A741$ operational amplifier; (f) BJT model.

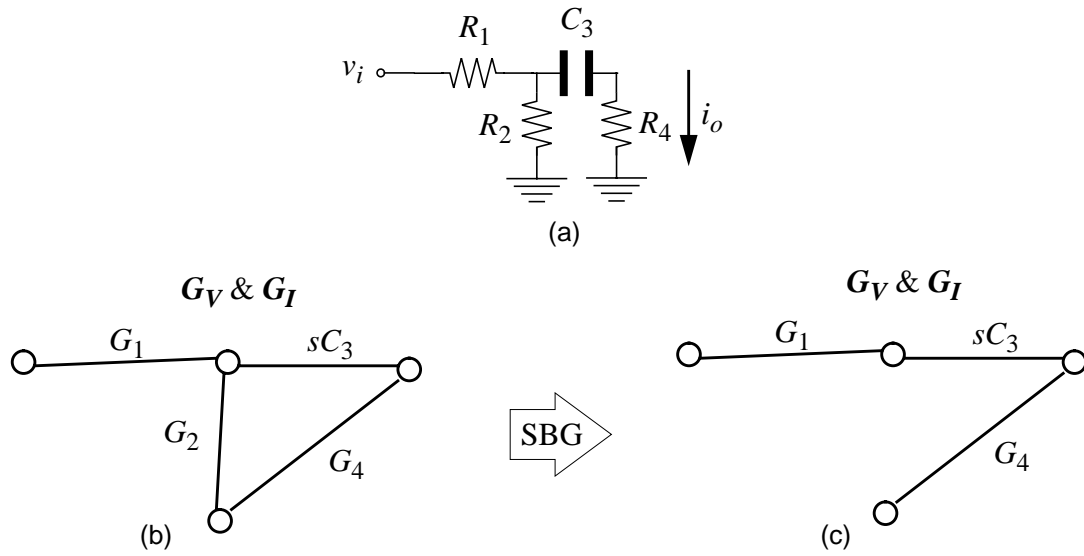


Figure 2 Illustrating an implementation of SBG: (a) Circuit; (b) Formulation of voltage and current graphs; (c) Simplified graphs.

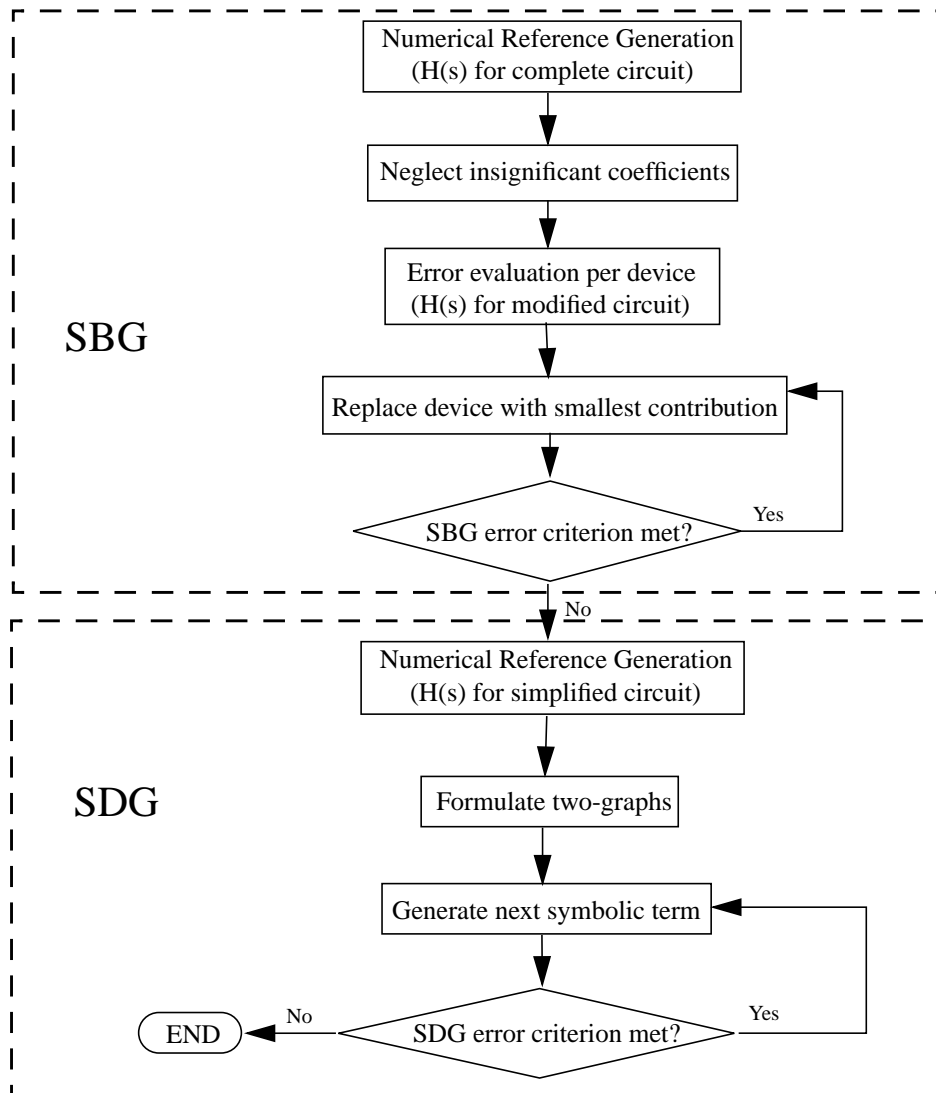


Figure 3 Symbolic analysis methodology for large analog circuits.

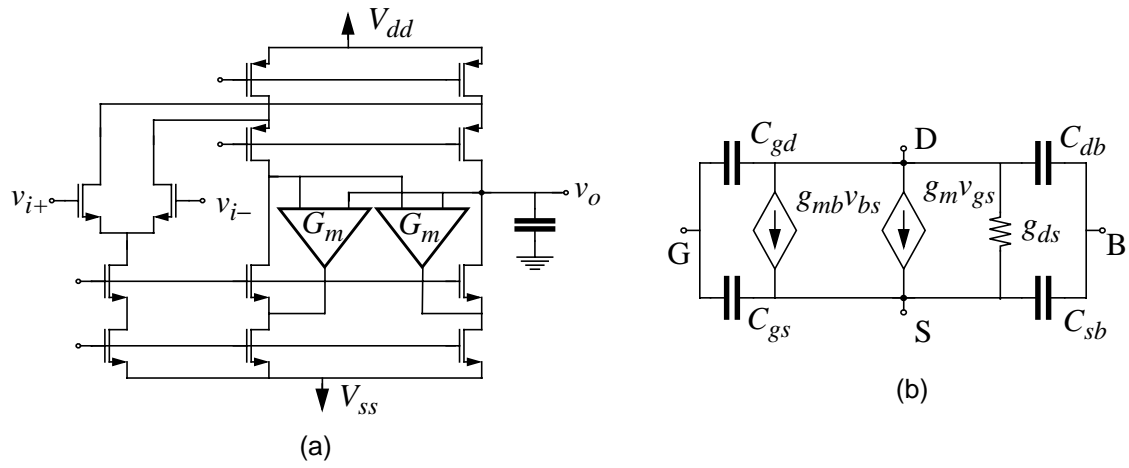


Figure 4 (a) Positive feedback OTA; (b) MOSFET model.

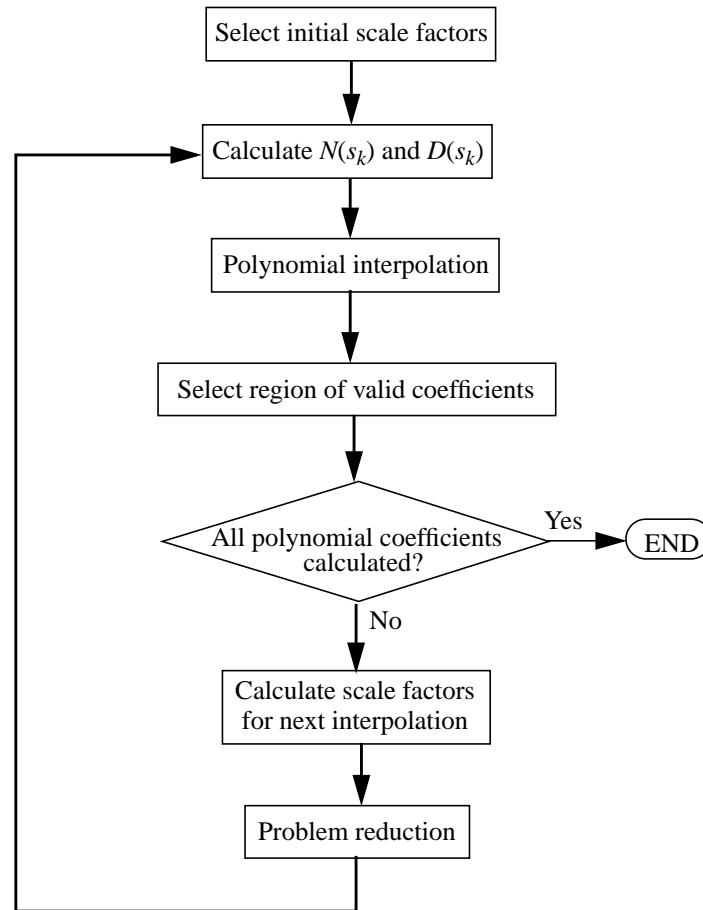


Figure 5 Proposed methodology for numerical reference generation for large analog circuits.

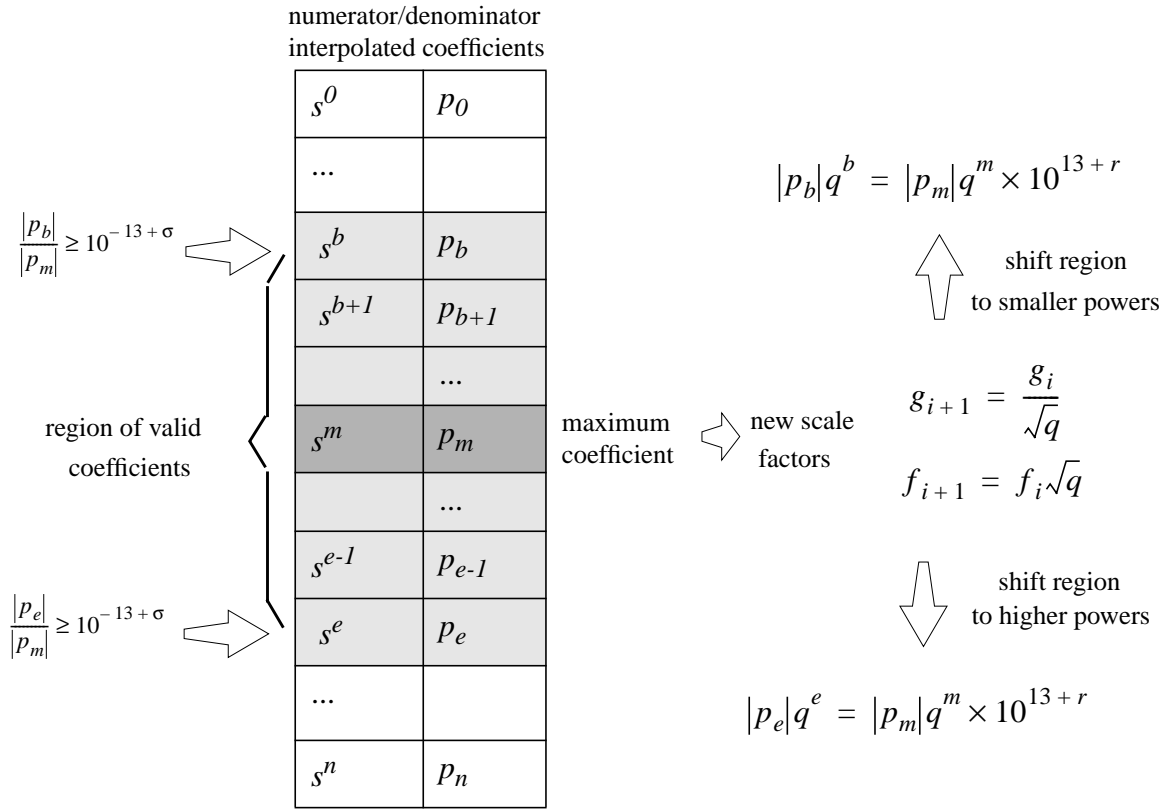


Figure 6 Illustrating the adaptive scaling mechanism at the i -th iteration.

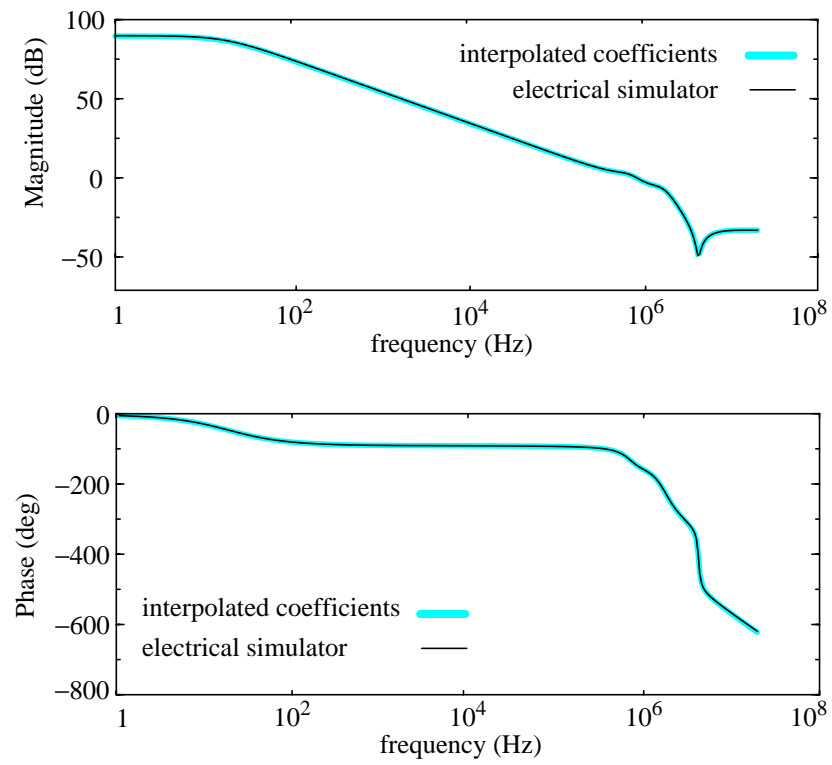


Figure 7 Bode diagrams of the voltage gain of the $\mu A741$ opamp using the interpolated coefficients and an electrical simulator.

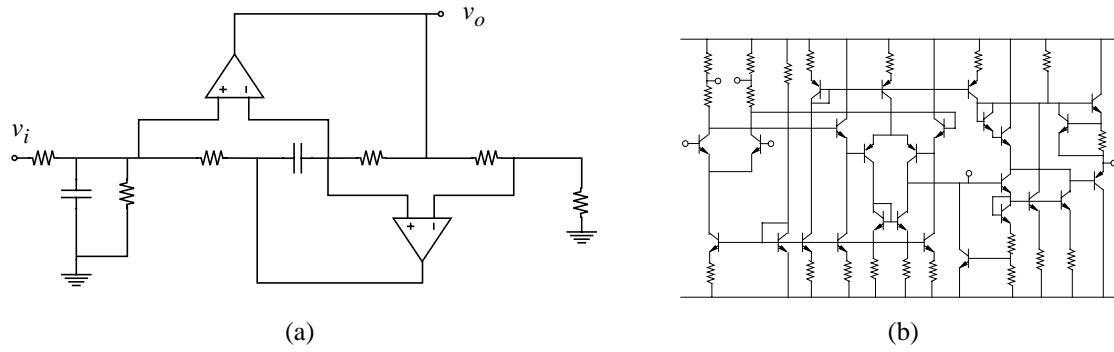


Figure 8 (a) Bandpass biquad; (b) $\mu A725$ opamp.

Table 1 Transfer function coefficients for the differential voltage gain of Fig. 4a using interpolation points on the unit circle.

s^i	Numerator coefficients	Denominator coefficients
s^0	$-5.8296e-25+j0.0$	$+8.9418e-30+j0.0$
s^1	$-1.5484e-33-j2.2958e-41$	$+3.8525e-36-j7.0064e-47$
s^2	$-2.5254e-41+j1.8367e-41$	$+2.3920e-43-j1.4013e-46$
s^3	$-5.5101e-41+j0.0$	$+1.0646e-43-j1.4013e-46$
s^4	$+7.3468e-41+j3.6734e-41$	$-8.4077e-46-j5.6051e-46$
s^5	$-4.5917e-41+j3.5695e-41$	$+2.1019e-45-j5.4751e-46$
s^6	$+5.5101e-41+j4.1326e-41$	$-4.2039e-46-j5.6051e-46$
s^7	$+1.8826e-40-j2.0203e-40$	$+1.0243e-43+j3.0828e-45$
s^8	$-1.1479e-40+j5.5101e-41$	$-1.8020e-43-j5.6051e-46$
s^9	$-1.7448e-40-j1.6530e-40$	$+6.8383e-43+j2.5223e-45$

Table 2 Denominator coefficients for the differential voltage gain of Fig. 4a using a frequency scale factor $f = 10^8$.

s^i	(a) Normalized coefficients	(b) Denormalized coefficients
s^0	8.94186e-30-j5.74242e-44	8.94186e-30-j5.74242e-44
s^1	3.85259e-28+j1.39200e-44	3.85259e-36+j1.39200e-52
s^2	4.20423e-27+j8.17887e-43	4.20423e-43+j8.17887e-59
s^3	1.31933e-27+j3.12804e-43	1.31933e-51+j3.12804e-67
s^4	1.69136e-28-j2.85774e-43	1.69136e-60-j2.85774e-75
s^5	1.09681e-29-j3.09269e-43	1.09681e-69-j3.09269e-83
s^6	3.58708e-31-j4.41609e-43	3.58708e-79-j4.41609e-91
s^7	4.72362e-33+j2.45089e-43	4.72362e-89+j2.45089e-99
s^8	9.71439e-42-j6.61683e-43	9.71439e-106-j6.61683e-107
s^9	-7.50537e-42-j2.6935e-42	-7.50537e-114-j2.6935e-114

Table 3 Valid voltage gain coefficients obtained from the first algorithm iteration on the μ A741 opamp in Fig. 1e.

s^i	Numerator coefficients		Denominator coefficients	
	Normalized	Denormalized	Normalized	Denormalized
s^0	-9.60926e+122	-5.58675e-86	-2.82408e+118	-1.6419e-90
s^1	-1.05987e+124	-2.10393e-91	-7.32222e+122	-1.45352e-92
s^2	-1.48757e+124	-1.00824e-97	-8.26327e+123	-5.60064e-98
s^3	-1.09256e+124	-2.52835e-104	-1.28095e+124	-2.96432e-104
s^4	-4.74222e+123	-3.74701e-111	-1.20867e+124	-9.55018e-111
s^5	-1.20465e+123	-3.24992e-118	-7.46903e+123	-2.015e-117
s^6	-1.7316e+122	-1.59502e-125	-3.17468e+123	-2.92428e-124
s^7	-1.17059e+121	-3.68155e-133	-9.73518e+122	-3.06176e-131
s^8	3.98904e+119	4.28355e-141	-2.19449e+122	-2.3565e-138
s^9	2.12204e+119	7.7803e-148	-3.61682e+121	-1.32608e-145
s^{10}	3.16408e+118	3.96094e-155	-4.2945e+120	-5.37606e-153
s^{11}	2.81205e+117	1.20194e-162	-3.61821e+119	-1.54651e-160
s^{12}	1.45161e+116	...	-2.13624e+118	-3.11759e-168
s^{13}	3.74942e+114	...	-8.7689e+116	...
...	
s^{48}	...			

Table 4 Valid voltage gain coefficients obtained from the second algorithm iteration on the μ A741 opamp in Fig. 1e.

s^i	Numerator coefficient		Denominator coefficient	
	Normalized	Denormalized	Normalized	Denormalized
...	
s^{11}	1.26823e+83	1.20194e-162	-1.6318e+85	-1.54651e-160
s^{12}	2.87085e+84	2.11845e-170	-4.22484e+86	-3.11759e-168
s^{13}	3.25114e+85	1.86795e-178	-7.60487e+87	-4.3694e-176
s^{14}	7.09905e+85	3.17579e-187	-9.50869e+88	-4.25375e-184
s^{15}	-2.41332e+87	-8.40596e-195	-8.31808e+89	-2.89732e-192
s^{16}	-3.10937e+88	-8.4327e-203	-5.16263e+90	-1.40012e-200
s^{17}	-1.93746e+89	-4.09119e-211	-2.31064e+91	-4.8792e-209
s^{18}	-7.5856e+89	-1.24718e-219	-7.57228e+91	-1.24499e-217
s^{19}	-2.03572e+90	-2.60601e-228	-1.84185e+92	-2.35783e-226
s^{20}	-3.91629e+90	-3.90351e-237	-3.36737e+92	-3.35638e-235
s^{21}	-5.55819e+90	-4.31356e-246	-4.68533e+92	-3.63616e-244
s^{22}	-5.94529e+90	-3.5925e-255	-5.02443e+92	-3.03607e-253
s^{23}	-4.87733e+90	-2.29471e-264	-4.20538e+92	-1.97857e-262
s^{24}	-3.11448e+90	-1.14091e-273	-2.78054e+92	-1.01858e-271
s^{25}	-1.56748e+90	-4.47086e-283	-1.46833e+92	-4.18806e-281
s^{26}	-6.28204e+89	-1.39512e-292	-6.25244e+91	-1.38854e-290
s^{27}	-2.02144e+89	-3.49537e-302	-2.1642e+91	-3.74221e-300
s^{28}	-5.2559e+88	-7.0762e-312	-6.12909e+90	-8.25182e-310
s^{29}	-1.10938e+88	-1.16293e-321	-1.4274e+90	-1.49631e-319
s^{30}	-1.9069e+87	-1.55641e-331	-2.7438e+89	-2.23949e-329
s^{31}	-2.67406e+86	-1.69938e-341	-4.36393e+88	-2.7733e-339
s^{32}	-3.06103e+85	-1.51463e-351	-5.74996e+87	-2.84515e-349
s^{33}	-2.85887e+84	-1.10143e-361	-6.27718e+86	-2.41839e-359
s^{34}	-2.17505e+83		-5.67206e+85	-1.70147e-369
s^{35}	-1.34446e+82		-4.2331e+84	
...	

Table 5 Valid voltage gain coefficients obtained from the third algorithm iteration on the μ A741 opamp in Fig. 1e.

s^i	Numerator coefficient		Denominator coefficient	
	Normalized	Denormalized	Normalized	Denormalized
...	
s^{31}	-9.54531e+100	-1.69938e-341	-1.55775e+103	-2.7733e-339
s^{32}	-1.31048e+102	-1.51463e-351	-2.46167e+104	-2.84515e-349
s^{33}	-1.46792e+103	-1.10143e-361	-3.2231e+105	-2.41839e-359
s^{34}	-1.33944e+104	-6.52456e-372	-3.49298e+106	-1.70147e-369
s^{35}	-9.93e+104	-3.14017e-382	-3.12651e+107	-9.88697e-380
s^{36}	-5.95853e+105	-1.22326e-392	-2.30357e+108	-4.72912e-390
s^{37}	-2.87924e+106	-3.83736e-403	-1.39051e+109	-1.85323e-400
s^{38}	-1.11283e+107	-9.62853e-414	-6.83395e+109	-5.91294e-411
s^{39}	-3.40957e+107	-1.91517e-424	-2.7127e+110	-1.52373e-421
s^{40}	-8.18174e+107	-2.98352e-435	-8.6082e+110	-3.13903e-432
s^{41}	-1.51217e+108	-3.57981e-446	-2.15541e+111	-5.10259e-443
s^{42}	-2.10129e+108	-3.2294e-457	-4.18739e+111	-6.43546e-454
s^{43}	-2.11603e+108	-2.11122e-468	-6.17333e+111	-6.1593e-465
s^{44}	-1.45292e+108	-9.41084e-480	-6.70084e+111	-4.34028e-476
s^{45}	-6.06159e+107	-2.54889e-491	-5.12766e+111	-2.15617e-487
s^{46}	-1.15315e+107	-3.14794e-503	-2.58524e+111	-7.05735e-499
s^{47}			-7.60583e+110	-1.34792e-510
s^{48}			-9.74782e+109	-1.1215e-522

Table 6 Valid denominator coefficients of the voltage gain of the bandpass biquad in Fig. 8a obtained from the (a) first, (b) second, (c) third and (d) fourth algorithm iteration.

s^i	(a) Denominator coefficients	
	Normalized	Denormalized
...	...	
s^{23}	-7.51056e+09	
s^{24}	-5.94246e+10	
s^{25}	-7.29364e+11	-5.2904e-395
...
s^{40}	-5.80151e+18	-3.26183e-513
s^{41}	-6.87954e+18	-1.70385e-521
s^{42}	-7.2521e+18	-7.91199e-530
s^{43}	-6.79335e+18	-3.2648e-538
...
s^{58}	-1.47487e+12	-3.23433e-670
s^{59}	-8.03012e+11	-1.96111e-679
...		

s^i	(b) Denominator coefficients	
	Normalized	Denormalized
s^0	-3.91876e+56	
s^1	-1.01152e+58	-4.94769e-230
s^2	-7.75231e+59	-4.44594e-235
...
s^8	-1.04737e+64	-1.56051e-272
s^9	-1.42046e+64	-2.48143e-279
s^{10}	-1.56628e+64	-3.2081e-286
s^{11}	-1.4507e+64	-3.48388e-293
s^{12}	-1.163e+64	-3.2747e-300
...
s^{23}	-2.94228e+60	-4.76909e-380
s^{24}	-8.74623e+59	-1.66218e-387
s^{25}	...calculated...	

s^i	(c) Denominator coefficients	
	Normalized	Denormalized
s^{59}	...calculated...	
s^{60}	-3.64197e+12	-1.05641e-688
...
s^{68}	-1.03089e+15	-1.183e-764
s^{69}	-1.26356e+15	-2.2963e-774
s^{70}	-1.39033e+15	-4.00138e-784
s^{71}	-1.37471e+15	-6.2656e-794
...
s^{86}	-4.32544e+09	-1.94841e-946
s^{87}	-7.99215e+08	-5.70129e-957
s^{88}	-1.41979e+08	-1.49098e-967
s^{89}	-1.94284e+07	
...

s^i	(d) Denominator coefficients	
	Normalized	Denormalized
s^{87}	...calculated...	
s^{88}	...calculated...	
s^{89}	-9.84964e+47	-3.47589e-978
...
s^{94}	-5.04429e+49	-3.76528e-1032
s^{95}	-7.39434e+49	-4.04545e-1043
s^{96}	-9.31988e+49	-3.73721e-1054
s^{97}	-1.00029e+50	-2.9399e-1065
s^{98}	-9.03506e+49	-1.9463e-1076
...
s^{104}	-3.93227e+47	-1.31324e-1145
s^{105}	-4.561e+46	-1.11642e-1157
s^{106}	-2.41567e+45	-4.33388e-1170

LIST OF FIGURES

- Figure 1** (a) BJT feedback amplifier; (b) low-frequency BJT model; (c) Miller operational amplifier; (d) MOSFET model; (e) mA741 operational amplifier; (f) BJT model.
- Figure 2** Illustrating an implementation of SBG: (a) Circuit; (b) Formulation of voltage and current graphs; (c) Simplified graphs.
- Figure 3** Symbolic analysis methodology for large analog circuits.
- Figure 4** (a) Positive feedback OTA; (b) MOSFET model.
- Figure 5** Proposed methodology for numerical reference generation for large analog circuits.
- Figure 6** Illustrating the adaptive scaling mechanism at the i -th iteration.
- Figure 7** Bode diagrams of the voltage gain of the μ A741 opamp using the interpolated coefficients and an electrical simulator.
- Figure 8** (a) Bandpass biquad; (b) μ A725 opamp.

LIST OF TABLES

- Table 1** Transfer function coefficients for the differential voltage gain of Fig. 4a using interpolation points on the unit circle.
- Table 2** Denominator coefficients for the differential voltage gain of Fig. 4a using a frequency scale factor.
- Table 3** Valid voltage gain coefficients obtained from the first algorithm iteration on the μ A741 opamp in Fig. 1e.
- Table 4** Valid voltage gain coefficients obtained from the second algorithm iteration on the μ A741 opamp in Fig. 1e.
- Table 5** Valid voltage gain coefficients obtained from the third algorithm iteration on the μ A741 opamp in Fig. 1e.
- Table 6** Valid denominator coefficients of the voltage gain of the bandpass biquad in Fig. 8a obtained from the (a) first, (b) second, (c) third and (d) fourth algorithm iteration.

FOOTNOTES

1. The number of terms for Fig. 1c was obtained using ASAP [3] while the lower bound of the number of terms for Fig. 1e was calculated using the theory presented in [4].

Symbolic Analysis of Large Analog Integrated Circuits: The Numerical Reference Generation Problem

*Francisco V. Fernández, Oscar Guerra, Juan D. Rodríguez-García and Angel
Rodríguez-Vázquez*

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA)
E-41012, Sevilla, Spain

***IEEE Transactions on Circuits and Systems - II,
Accepted for publication to appear in 1998***

© 1998 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reported without the explicit permission of the copyright holder.

Symbolic Analysis of Large Analog Integrated Circuits: The Numerical Reference Generation Problem

*Francisco V. Fernández, Oscar Guerra, Juan D. Rodríguez-García and Angel
Rodríguez-Vázquez*

Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica
E-41012-Sevilla, SPAIN

Footnote

This work has been supported by the EEC ESPRIT Program in the Framework of the Project #21812 (AMADEUS) and the Spanish C.I.C.Y.T. under contract TIC97-0580.

Abstract

Symbolic analysis potentialities for gaining circuit insight and for efficient repetitive evaluations have been limited by the exponential increase of formula complexity with the circuit size. This drawback has began to be solved by the introduction of simplification before and during generation techniques. An appropriate error control in both involves the generation of a numerical reference, which implies the calculation of network functions in the complex frequency variable. The polynomial interpolation method, traditionally used for this task, is analyzed in detail, its limitations for large circuit analysis are pointed out, and an adaptive scaling mechanism is proposed to meet the efficiency and accuracy requirements imposed by the new simplification methodologies.

I. INTRODUCTION

Symbolic circuit analysis refers to the calculation of network functions where the complex frequency and all or part of the circuit parameters are symbols. These functions are typically given in the form:

$$H(s, \mathbf{x}) = \frac{\sum_{i=0}^N s^i f_i(\mathbf{x})}{\sum_{j=0}^M s^j g_j(\mathbf{x})} \quad (1)$$

where $f_i(\mathbf{x})$ and $g_j(\mathbf{x})$ are sums of products of the symbolic parameters $\mathbf{x}^T = \{x_1, x_2, \dots, x_Q\}$. See for instance [1] and [2] for an actualized review of techniques and applications of symbolic analysis.

Plain symbolic analysis suffers from a tremendous increase of expression complexity with the circuit size. Consider for illustration's sake the circuits in Fig. 1. The DC voltage gain of Fig. 1a using the model in Fig. 1b is:

$$\frac{v_o}{v_i} = [g_{m1}g_{m2}r_{\pi1}r_{\pi2}R_3R_L(R_1 + R_2) + R_1R_L(R_3 + r_{\pi2})g_{m1}r_{\pi1} + R_1R_L(R_3 + r_{\pi2})] / [g_{m1}g_{m2}r_{\pi1}r_{\pi2}R_3R_LR_1 + g_{m1}r_{\pi1}(R_2 + R_L)(R_3 + r_{\pi2})R_1 + (R_2 + R_L)(R_3 + r_{\pi2})(R_1 + r_{\pi1}) + (R_3 + r_{\pi2})r_{\pi1}R_1] \quad (2)$$

which contains 21 terms; this number raises to 8616 for the Miller opamp in Fig. 1c using the model in Fig. 1d, and is well above 10^{16} for the $\mu A741$ opamp in Fig. 1e using the model in Fig. 1f¹. Given this exponential increase of the term count with the number of elements in the circuit model, symbolic expression *simplification* has been recognized to be essential for both: *formula interpretation* by human designers and computer manipulation for *repetitive evaluations* in design automation applications [1]. For instance, elimination of the least significant terms in (2) leads to $\frac{v_o}{v_i} \approx \frac{(R_1 + R_2)}{R_1}$ which is a much more interpretable expression.

Conventional simplification approaches first calculate the *complete* symbolic expression, and then simplify it by eliminating insignificant terms or sub-expressions, based on numerical estimates of the symbolic parameters—commonly called Simplification After Generation (SAG). Consequently, most of the resources employed to generate the pruned terms are wasted. Besides, although this is a feasible approach for circuits like those of Fig. 1a and c (in general,

1. The number of terms for Fig. 1c was obtained using ASAP [3] while the lower bound of the number of terms for Fig. 1e was calculated using the theory presented in [4].

for circuits with less than around 50 symbols), it is unfeasible for circuits like that in Fig. 1e, as no computer has enough memory to handle such a huge number of symbolic terms, on the one hand, and the time needed to generate them would not be affordable, on the other. These larger circuits have to be analyzed by using the newest approaches: Simplification During Generation (SDG) and Simplification Before Generation (SBG). This paper deals with a basic ingredient of these new techniques, namely the generation of a numerical reference to evaluate the errors in the simplification process. Based on a brief description of our implementation of SBG and SDG (Section II), Section III addresses the generation of this reference, describing the problems arising when handling medium and large size analog integrated circuits and introducing new algorithms for its efficient calculation. Experimental results are shown in Section IV.

II. THE APPROXIMATION METHODOLOGY

A. Simplification During Generation

SDG techniques start from some formulation of the network equations and solve them trying to directly generate the simplified expression. In our SDG approach, symbolic terms are generated in decreasing order of magnitude until the generated terms represent a significant fraction of the complete expression.

The first reliable algorithms capable of efficiently generating terms in decreasing order of magnitude [5]-[8] were based on the *two-graph* method [9]. The computation of the simplified coefficients of s^k reduces to the following problem: “Given the voltage graph G_V and the current graph G_I of a circuit with n nodes, enumerate subsets of $(n - 1)$ branches in decreasing order of magnitude such that: (a) form a spanning tree in G_V ; (b) form a spanning tree in G_I ; and, (c) contain k capacitances and $(n - k - 1)$ (trans)conductances”. This problem can be formulated in terms of *matroids* [10]. Each condition (a)–(c) above is mapped into a matroid and the problem of generation of common spanning trees in order is mapped into a weighted matroid intersection problem. The algorithms in [5]-[8] calculate the intersection of two matroids among (a)-(c) [11], and then check if it intersects the third matroid. Although the intersection problem of three general matroids is nonpolynomial-hard, [12], [13] have reported the first algorithm able to solve it by exploiting the characteristics of the three particular matroids at hand. These algorithms have made feasible the analysis of large circuits like the $\mu A741$ opamp in a few tens of seconds.

An important ingredient of SDG is the error criterion used to stop the generation of terms.

Consider that $h_k(\mathbf{x}) = \sum_{l=1}^T h_{kl}(\mathbf{x})$ represents either $f_i(\mathbf{x})$ or $g_j(\mathbf{x})$ in (1). The P most significant

terms are generated in $h_k(\mathbf{x})$ until the sum of the generated terms represents a given fraction of the total magnitude of the coefficient,

$$\left| h_k(\mathbf{x}_o) - \sum_{l=1}^P h_{kl}(\mathbf{x}_o) \right| < \varepsilon_k |h_k(\mathbf{x}_o)| \quad (3)$$

where \mathbf{x}_o represents a design point of the circuit parameters and ε_k is an error control parameter which is obtained by backpropagation from maximum magnitude and phase error specifications. As shown in (3), the total magnitude of each circuit coefficient, $h_k(\mathbf{x}_o)$, must be known a priori; however, the fully symbolic expression is not available for such calculation. Hence, an efficient technique able to calculate (1) with only s as symbolic variable is needed. The problems arising in this calculation when handling large circuits are addressed in Section III.

Once an approximated expression $H_{ap}(s)$ has been calculated, the maximum magnitude and phase errors with respect to the exact expression $H_{ex}(s)$ in a given frequency range can be obtained from:

$$\varepsilon_{|H|} = \frac{|H_{ex}(j\omega)| - |H_{ap}(j\omega)|}{|H_{ex}(j\omega)|} = 1 - \frac{\sqrt{\frac{N_{apr}^2 + N_{api}^2}{D_{apr}^2 + D_{api}^2}}}{\sqrt{\frac{N_{exr}^2 + N_{exi}^2}{D_{exr}^2 + D_{exi}^2}}} \quad (4)$$

$$\Delta\phi_H = \angle H_{ex}(j\omega) - \angle H_{ap}(j\omega) = \text{atan} \frac{N_{exi}}{N_{exr}} - \text{atan} \frac{D_{exi}}{D_{exr}} - \text{atan} \frac{N_{api}}{N_{apr}} + \text{atan} \frac{D_{api}}{D_{apr}}$$

where

$$\begin{aligned} H_{ap}(j\omega) &= \frac{N_{ap}(j\omega)}{D_{ap}(j\omega)} = \frac{N_{apr}(\omega) + jN_{api}(\omega)}{D_{apr}(\omega) + jD_{api}(\omega)} \\ H_{ex}(j\omega) &= \frac{N_{ex}(j\omega)}{D_{ex}(j\omega)} = \frac{N_{exr}(\omega) + jN_{exi}(\omega)}{D_{exr}(\omega) + jD_{exi}(\omega)} \end{aligned} \quad (5)$$

The application of interval analysis techniques [14] to (4) to evaluate the maximum magnitude and phase errors in a given frequency range usually yields overly conservative estimates of those maxima. Therefore, interval analysis techniques are applied to the derivatives of (4) to delimit frequency subranges in which the maximum magnitude and phase errors occur. Then, the frequency points for which the maximum magnitude or phase error occurs in those frequency subranges are easily calculated using the Newton-Raphson method.

B. Simplification Before Generation

SBG performs the approximation during the set-up of the network equations by eliminating matrix entries, graph branches, etc. Then, the reduced matrix or graph is much easier to solve. Our SBG approach takes place at the graph level, replacing those elements, whose contribution (appropriately measured) to the network function is negligible, with a zero-admittance or zero-impedance element. This is illustrated with the simple example in Fig. 2. Assuming that G_2 is evaluated to have the smallest contribution to the network function i_o/v_i , it can be deleted from the voltage and current graphs. The network function for the simplified circuit is

$$\frac{i_o}{v_i} = \frac{C_3 s}{1 + (R_1 C_3 + R_4 C_3) s} \quad (6)$$

which is significantly less complex than that resulting for the original graphs:

$$\frac{i_o}{v_i} = \frac{R_2 C_3 s}{(R_1 + R_2) + (R_1 R_2 C_3 + R_1 R_4 C_3 + R_2 R_4 C_3) s} \quad (7)$$

The same deletion/contraction operation is repeated for the next element with smallest contribution, and so on. The reduction in formula complexity is more significant for larger circuits.

Reported approaches evaluate the influence of the elimination of matrix entries [15], [16] or graph branches [7] at a single or at a finite number of sample frequency points, and hence do not guarantee accuracy at other frequency points. To solve this problem we evaluate each element contribution by comparing the network function of the complete circuit and that of a modified circuit in which the element has been deleted/contracted. This implies calculating the network function as a function of s for each deletion/contraction. The polynomial interpolation method, which is considered to be the most efficient one to perform this task, is analyzed in detail in Section III. Detection of the maximum magnitude and phase errors induced by a device replacement is performed as described in Section IIA.

However, even the techniques presented in Section III to improve the efficiency of this method are not sufficient for a repetitive application in SBG. But, it usually turns out that many numerator and denominator coefficients do not have a significant contribution in the frequency range of interest. A large error in those coefficients is unimportant and, hence, they can be neglected. For instance, once the coefficients of the voltage gain of the $\mu A741$ opamp have been calculated, all those which only become significant above 10MHz can be neglected as the opamp will never be operated at such frequency. This operation drastically reduces the cost of

subsequent network function calculations as the polynomial interpolation cost grows with the number of network function coefficients.

The flow diagram of the complete symbolic analysis methodology including SBG and SDG is shown in Fig. 3. It must be noticed that the SBG step makes the SDG task much more effective but it is not essential, that is, SDG can do the work without a previous SBG step.

III. NETWORK FUNCTIONS IN s

Section II has shown that error evaluation in both, SBG and SDG techniques, requires repetitive calculation of network functions with s as the unique symbolic variable for complete or reduced circuit models. This establishes how important is to develop efficient calculation techniques of network functions of medium and large size circuits. The polynomial interpolation method is considered to be one of the most efficient techniques to perform this task [9],[17].

A. Background on polynomial interpolation

The polynomial interpolation starts from the fact that the coefficients of a n -th order polynomial,

$$P(s) = p_0 + p_1s + p_2s^2 + \dots + p_ns^n \quad (8)$$

can be obtained from the polynomial values at $(n+1)$ distinct points $P(s_k)$. If these values can be calculated, then the following matrix equation can be formulated:

$$\begin{bmatrix} 1 & s_0 & s_0^2 & \dots & s_0^n \\ 1 & s_1 & s_1^2 & \dots & s_1^n \\ \dots & & & & \\ 1 & s_n & s_n^2 & \dots & s_n^n \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_n \end{bmatrix} = \begin{bmatrix} P(s_0) \\ P(s_1) \\ \dots \\ P(s_n) \end{bmatrix} \quad (9)$$

The matrix in (9) is nonsingular and hence (9) has always a unique solution. Such solution is the set of polynomial coefficients, p_i , in (8).

It has been shown that the use of $K \geq n + 1$ equally-spaced interpolation points in the unit circle gives the best results concerning numerical accuracy and stability [9], [17]. Once the values of (8) at all these points $P(s_k)$ are known, the polynomial coefficients can be obtained through the *Discrete Fourier Transform* (DFT),

$$\hat{p}_i = \frac{1}{K} \sum_{k=0}^{K-1} P(s_k) e^{-\frac{2\pi i k}{K}} \quad i = 0, 1, \dots, K-1 \quad (10)$$

where

$$\hat{p}_i = \begin{cases} p_i & \text{for } i \leq n \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The number of interpolation points, K , should be at least $(n+1)$, but in most cases, like that we are dealing with, the polynomial order n is not known beforehand. Hence, an upper estimate on K must be done, and (10) should be identically 0 for those coefficients over the n -th power.

Our objective is not the calculation of a polynomial but a network function, which is given by the ratio of two polynomials. Therefore, the polynomial interpolation method is applicable to our problem once the values of the numerator $N(s_k)$ and denominator $D(s_k)$ at the different interpolation points are known. In order to calculate $N(s_k)$ and $D(s_k)$ assume that an appropriate formulation method, i.e. modified nodal analysis, has been applied on the circuit so that the network equations can be written as:

$$\mathbf{Y}_{MNA} \mathbf{X} = \mathbf{E} \quad (12)$$

where \mathbf{Y}_{MNA} is the modified nodal matrix, \mathbf{X} contains nodal voltages and auxiliary currents, and \mathbf{E} accounts for the influence of the independent sources. Once any frequency-dependent element in \mathbf{Y}_{MNA} is evaluated at the interpolation point $s=s_k$, the value of the network function

$$H(s_k) = \frac{N(s_k)}{D(s_k)} \quad (13)$$

can be obtained by applying LU decomposition and backsubstitution to (12). The denominator of the network function is easily obtained as:

$$D(s_k) = |\mathbf{Y}_{MNA}(s_k)| \quad (14)$$

and the numerator $N(s_k)$ is easily obtained from (13) and (14):

$$N(s_k) = H(s_k) \cdot D(s_k) \quad (15)$$

B. Introducing scaling

One major problem in polynomial interpolation applied to analog integrated circuits is the

dramatic effect of round-off errors, due to the finite precision arithmetics of computers. The calculation of the differential voltage gain as a function of s in the positive feedback OTA of Fig. 4a with the transistor model in Fig. 4b shows these problems. The order of this network function is unknown a priori, but an upper bound of the order can be estimated at 9; hence 10 interpolation points are used. The interpolated numerator and denominator coefficients when using interpolation points located at the unit circle are given in Table 1.

Polynomial coefficients must be real, but, as shown in Table 1, many interpolated coefficients have a non-zero imaginary component. This is due to the round-off errors, which avoid perfect cancellations of the imaginary parts in the DFT. The values of these imaginary components give us an idea of the numerical noise level induced by the finite number of bits available in a digital computer to represent the floating point numbers. The real and imaginary parts of the interpolated coefficients in Table 1, except the two shadowed ones, are of the same order of magnitude; therefore, the actual value of those coefficients is not obtained as it is below the numerical error level. Also, as indicated by (11) the zero coefficients would indicate the actual polynomial order, but as shown in Table 1 the zero coefficients are not detected now.

The numerical error level in the polynomial interpolation depends on the coefficient p_i in (8) having the largest absolute value. This error level is about $10^{-13} \times \max_i |p_i|$ in a computer with 16-decimal-digit accuracy [17],[18]. The spread of values between the maximum and minimum coefficient should be well below this error to ensure numerical accuracy of the calculated coefficients:

$$\frac{\min_i |\text{coefficient}_i|}{\max_i |\text{coefficient}_i|} \gg 10^{-13} \quad (16)$$

Hence, it is not difficult to see that the second and higher order coefficients in Table 1 are not valid.

Each polynomial coefficient in typical analog integrated circuits is a sum-of-products of admittances: (trans)conductances and capacitances. Therefore, the coefficient of s^i has one more (trans)conductance and one less capacitance in each term than the coefficient of s^{i+1} . Taking into account the typical magnitudes of (trans)conductances and capacitances in analog circuits we can expect an extremely large spread of coefficient values. In order to reduce it, the complex frequency variable (equivalently the capacitor values) should be scaled before performing the polynomial interpolation on the unit circle [18]. Also, this suggests conductance scaling as another alternative.

The effect of conductance and frequency scaling on the relative value of the coefficients is analogous. Assume that g is the conductance scale factor and f is the frequency scale factor, then the polynomial with normalized coefficients p'_i is:

$$P(s) = \sum_{i=0}^n p'_i s^i = \sum_{i=0}^n p_i f^i g^{L-i-1} s^i = g^{L-1} \sum_{i=0}^n p_i \left(\frac{f}{g}\right)^i s^i \quad (17)$$

where L is the number of AC nodes in the circuit. We can observe that increasing the frequency scale factor has the same effect as decreasing the conductance scale factor.

For illustration's sake, Table 2a shows the normalized denominator coefficients, which were obtained using a frequency scale factor $f = 10^8$. The shadowed coefficients are well above the error level $10^{-13} \times \max_i |p_i|$, where the maximum coefficient is the second-order one, dark-shadowed in Table 2. Therefore, all shadowed coefficients can be considered to be correct. Table 2b shows the denominator coefficients after frequency denormalization. An equivalent result is obtained for the numerator.

C. Adaptive scaling

A major problem is that the appropriate scale factor is not known a priori. Moreover, for larger circuits (and correspondingly higher order network functions) no scale factor can be found that keeps the spread of coefficient values under the error level.

The solution to this problem must clearly go through the use of several scale factors. [18] proposed the selection of several frequency scale factors to obtain sets of valid coefficients. However, Section II has shown that SBG and SDG methodologies require extremely efficient techniques for the numerical reference generation. Therefore, the tentative proposal and test of many scale factors is not a viable solution.

Our objective is to make a minimum number of interpolations. The proposed solution is an iterative mechanism which performs successive interpolations using an adaptive scaling mechanism as schematically shown in the flow diagram in Fig. 5. The algorithm operation at the i -th operation is illustrated in Fig. 6.

After each polynomial interpolation a region of valid coefficients is selected. Assume that p_k 's are the interpolated coefficients (normalized by the scale factors g_i and f_i) at iteration i and p_m is the polynomial coefficient with the largest absolute value (dark-shadowed in Fig. 6). If coefficients with σ significant digits need to be calculated, then all coefficients p_k which prior to denormalization are

$$|p_k| < 10^{-13+\sigma} \times \max_i |p_i| = 10^{-13+\sigma} \times |p_m| \quad (18)$$

must be neglected. Assume that p_b and p_e limit the region of coefficients which meet the condition in (18). All coefficients in that region are considered to be valid and marked as calculated (light-shadowed in Fig. 6). Based on these results two new scale factors f_{i+1} and g_{i+1} are calculated for the following polynomial interpolation as follows:

$$g_{i+1} = \frac{g_i}{\sqrt{q}} \quad f_{i+1} = f_i \sqrt{q} \quad (19)$$

where

$$|p_b|q^b = |p_m|q^m \times 10^{13+r} \quad (20)$$

to shift the region of valid coefficients to smaller powers of s , being p_b and p_m the first and maximum coefficients within the last valid region respectively, and r a tuning factor; and

$$|p_e|q^e = |p_m|q^m \times 10^{13+r} \quad (21)$$

to shift the region of valid coefficients to higher powers of s , being p_e the last coefficient of the previous valid region.

The objective of the tuning factor r in (20)-(21) is that the region of valid coefficients at the $(i+1)$ -th iteration has the smallest overlapping with the region at the i -th iteration. To clarify the effect of r , consider for instance the use of (19) and (21) to shift the region of valid coefficients to higher powers of s . $|p_e|$ is slightly larger than $10^{-13+\sigma} \times |p_m|$. By choosing $r = \sigma$ the scale factor q given by (21) makes that the lower value for the region of valid coefficients in the subsequent iteration is approximately $|p_e|$. Since this is an approximation and to avoid that one coefficient remains uncalculated between the two consecutive valid regions the tuning factor r is heuristically chosen smaller: $r = \sigma - 1$. A more conservative selection, $r \ll \sigma$, is not advisable because that would make many coefficients to be redundantly calculated in subsequent interpolations with the corresponding extra computational effort

The scale factors for the first interpolation are calculated as a function of the mean value of capacitances and (trans)conductances. The objective of these heuristics is to first generate the widest region of valid coefficients.

If between two consecutive valid regions, with scale factors f_j, g_j and f_l, g_l , some incorrect coefficients remain, then new scale factors f_{i+1}, g_{i+1} , are calculated as follows,

$$f_{i+1} = f_j \cdot f_l \qquad g_{i+1} = g_j \cdot g_l \qquad (22)$$

Notice that in the algorithm above simultaneous scaling of both, frequency and conductance is used. This technique is used to avoid using too large ($> \sim 10^{18}$) frequency or conductance scale factors. These high values occasionally occur when using a single scale factor and are responsible for an increase of the error in the calculation of numerator and denominator of the transfer function at the interpolation points.

D. Problem reduction mechanism

In each polynomial interpolation the computational effort depends on the number of interpolation frequencies needed. The problem complexity can be reduced at subsequent iterations of previous algorithm, once the coefficients of the highest or smallest powers of s have been calculated. Assume the coefficients $p_0 \dots p_{k-1}$ and $p_{l+1} \dots p_n$ have already been calculated, then the polynomial is transformed as follows,

$$P'(s) = p_k + \dots + p_l \cdot s^{l-k} = \frac{P(s) - \sum_{i=0}^{k-1} p_i \cdot s^i - \sum_{i=l+1}^n p_i \cdot s^i}{s^k} \qquad (23)$$

The new polynomial contains the coefficients that still have to be calculated and needs only $l-k+1$ interpolation points. This simple operation drastically reduces the computation time at subsequent iterations.

IV. EXPERIMENTAL RESULTS

The proposed algorithm is applied in this section to two examples: the analysis of the $\mu A741$ opamp and a bandpass biquad described at the transistor level.

A. The $\mu A741$ opamp

Consider the voltage gain of the $\mu A741$ opamp in Fig. 1e with the small-signal BJT model of Fig. 1f. The results of the polynomial interpolation with the first frequency and conductance scale factors are partially shown in Table 3. The imaginary parts of the coefficients have been omitted from Table 3 as they originate from the numerical noise in the DFT. According to (18), the region of valid denominator coefficients is determined from the coefficient with largest normalized absolute value (dark-shadowed in Table 3): p_3 . If 6 significant digits are desired all denominator coefficients larger than

$$10^{-13+6} \times 1.28095 \times 10^{124} = 1.28095 \times 10^{117} \quad (24)$$

are considered correct. That means that the region of valid denominator coefficients extends from p_0 to p_{12} (light-shadowed). The remaining coefficients are not shown in Table 3 as it lacks interest. As indicated by (13)-(15) numerator and denominator coefficients are obtained simultaneously with a minimum extra cost. Hence, a region of valid numerator coefficients is also determined analogously.

The results of the first interpolation are used to calculate new scale factors f_2 and g_2 able to provide a region of valid coefficients of higher powers of s . For this, (19) and (21) are applied using $p_e = p_{12}$ and $p_m = p_3$. The problem reduction given by (23) allows the use of 13 less interpolation points in the next polynomial interpolation. The generated coefficients are shown in Table 4. The maximum absolute value coefficient, p_{22} , and the application of (18) delimits again the region of valid coefficients, which, as shown in Table 4, has shifted to the region between the 12-th and the 34-th coefficient of the denominator. It can be seen that the overlap between the valid region in Table 3 and Table 4 reduces to one coefficient in the denominator and there is no overlap in the numerator.

Again, (19) and (21) are applied to the coefficients in Table 4 to get a new set of scale factors for higher order coefficients. The problem reduction mechanism reduces in 22 less interpolation points for the third (last) iteration of the algorithm. The remaining coefficients, obtained in the third polynomial interpolation, are shown in Table 5. In this case there is an overlap of two coefficients in numerator and denominator.

The CPU time to get the results in this example was 3.9s for the first iteration, 2.3s for the second one and 0.9s for the third one (measured on a SPARC Station 10). The decrease in the number of interpolation points due to the problem reduction mechanism is clearly reflected in a CPU time reduction at subsequent iterations.

The accuracy of the results obtained in this example is demonstrated through the comparison of the Bode diagrams obtained from the interpolation of numerator and denominator of the voltage gain of $\mu A741$ and those obtained through a commercial electrical simulator, which are shown in Fig. 7. A perfect matching appears in all the frequency range.

B. Bandpass biquad

As a second example consider the bandpass biquad in Fig. 8a with the opamps described at the transistor level, as shown in Fig. 8b. The small-signal model used for the bipolar transistors was the same as for the previous example, shown in Fig. 1f. For limited space reasons and

without loss of generality we will limit ourselves to the calculation of the denominator of the voltage gain of the biquad.

The polynomial interpolation with the first set of frequency and conductance scale factors provides a region of valid coefficients which extends from the 25-th to the 59-th coefficient, as shown in Table 6a. To shift the region of valid coefficients to smaller powers of the frequency, (19) and (20) are applied to the results of the first interpolation. No problem reduction can be performed at this iteration. However, for the scale factors used the coefficients above the 59-th one are smaller than the error level. With the new scale factors shifting the valid region to smaller powers of s , the influence of the coefficients above the 59-th one on the polynomial value will be still smaller, and, hence, can be neglected. Neglecting these high order coefficients is useful because it allows to handle the polynomial as of smaller order, reducing in this way the number of points needed in the interpolation.

The polynomial interpolation with the new set of scale factors gives a region of valid coefficients which extends from the second to the 24-th coefficient and is shown in Table 6b. The calculation of the first coefficient, which is under the numerical error level, does not need an additional polynomial interpolation but a single LU decomposition with no frequency-dependent element in the circuit.

Then, the first 60 denominator coefficients are available, the problem is reduced, and, hence, 60 less interpolation points are needed at the following iteration. Now, the region of valid coefficients must be shifted to higher powers of s ; so, (19) and (21) are applied to the results of the first iteration of the algorithm. A new polynomial interpolation gives the results shown in Table 6c, where the region of valid coefficients is light-shadowed in Table 6c and its limits have been determined by (18). Again, the results of this interpolation are used to calculate new scale factors to shift the region of valid coefficients to higher powers of s , and to reduce the number of interpolation points at the following iteration. A new polynomial interpolation gives finally the remaining denominator coefficients, shown in Table 6d.

The CPU time spent to get the results shown in Table 6 is 30s. This time rises to 80s in case the problem reduction mechanism is not used. It could be argued that the CPU time obtained in these examples is acceptable for SDG where the network function in s must be calculated only once, while it is still too high for SBG where the polynomial interpolation or network function calculation might need to be calculated hundredths of times. This is not commonly true as for real circuits the results of the first network function calculation can be used to neglect a large number of coefficients for the frequency range in which we are inter-

ested. That means that the number of interpolation points and, hence, the CPU time, is drastically reduced for the following executions of the algorithm.

V. CONCLUSIONS

This paper has addressed the problems arising in the calculation of the numerical references, essential for an accurate error control in the proposed Simplification Before and During Generation approach for symbolic analysis of large analog circuits. The proposed algorithm is based on the polynomial interpolation method and incorporates simultaneous frequency and conductance scaling, an adaptive updating of the scale factors, and a problem reduction mechanism to speed up the generation of the numerical references. The experimental results obtained with large real-life circuits demonstrate the practical applicability of the techniques introduced in the paper.

VI. REFERENCES

- [1] F.V. Fernández, A. Rodríguez-Vázquez, J.L. Huertas and G. Gielen, eds., *Symbolic Analysis Techniques and Applications to Analog Design Automation*. Piscataway, NJ: IEEE Press, 1998.
- [2] G. Gielen, P. Wambacq and W. Sansen, "Symbolic analysis methods and applications for analog circuits: A tutorial overview," *Proc. of the IEEE*, vol. 82, no. 2, pp. 287-304, February 1994.
- [3] F. V. Fernández, A. Rodríguez-Vázquez, J. D. Martín, and J. L. Huertas, "Formula approximation for flat and hierarchical symbolic analysis," *Analog Integrated Circuits and Signal Processing*, vol. 3, no. 1, pp. 43-58, January 1993.
- [4] M. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms*. New York: John Wiley and Sons, 1981.
- [5] F.V. Fernández, P. Wambacq, G. Gielen, A. Rodríguez-Vázquez and W. Sansen, "Symbolic analysis of large analog integrated circuits by approximation during expression generation," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. CAD, pp. 25-28, 1994.
- [6] P. Wambacq, F.V. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez, "Efficient symbolic computation of approximated small-signal characteristics of analog integrated circuits," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 327-330, March 1995.
- [7] Q. Yu and C. Sechen, "Approximate symbolic analysis of large analog integrated circuits," *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 664-671, 1994.
- [8] Q. Yu and C. Sechen, "Efficient approximation of symbolic network functions using matroid intersection algorithms," *Proc. IEEE Int. Symp. Circuits Systems*, pp. 2088-2091, 1995.
- [9] P.M. Lin, *Symbolic Network Analysis*. Amsterdam: Elsevier, 1991.
- [10] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.
- [11] P.M. Camerini and H.W. Hamacher, "Intersection of two matroids: (condensed) border graph and ranking", *SIAM J. Discrete Mathematics*, vol. 2, pp. 16-27, February 1989.

- [12] M. Galán, I. García-Vargas, F.V. Fernández and A. Rodríguez-Vázquez, "A new matroid intersection algorithm for symbolic large circuit analysis," *Proc. Workshop on Symbolic Methods and Applications to Circuit Design*, Leuven, Belgium, 1996.
- [13] M. Galán, F.V. Fernández and A. Rodríguez-Vázquez, "Comparison of matroid intersection algorithms for large circuit analysis," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1784-1787, 1997.
- [14] R. E. Moore, *Methods and Applications of Interval Analysis*. Studies in Applied Mathematics, Philadelphia, 1979.
- [15] Jer-Jaw Hsu and C. Sechen, "Fully symbolic analysis of large analog integrated circuits," *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 21.4.1–21.4.4, 1994.
- [16] R. Sommer, E. Hennig, G. Droge, and E.-H. Horneber, "Equation-based symbolic approximation by matrix reduction with quantitative error prediction," *Alta Frequenza*, vol. 5, no. 6, pp. 317–325, November 1993.
- [17] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, 1994.
- [18] K. Singhal and J. Vlach, "Generation of immittance functions in symbolic form for lumped distributed active networks," *IEEE Trans. Circuits and Systems*, vol. CAS-21, no. 1, pp. 57-67, January 1974.

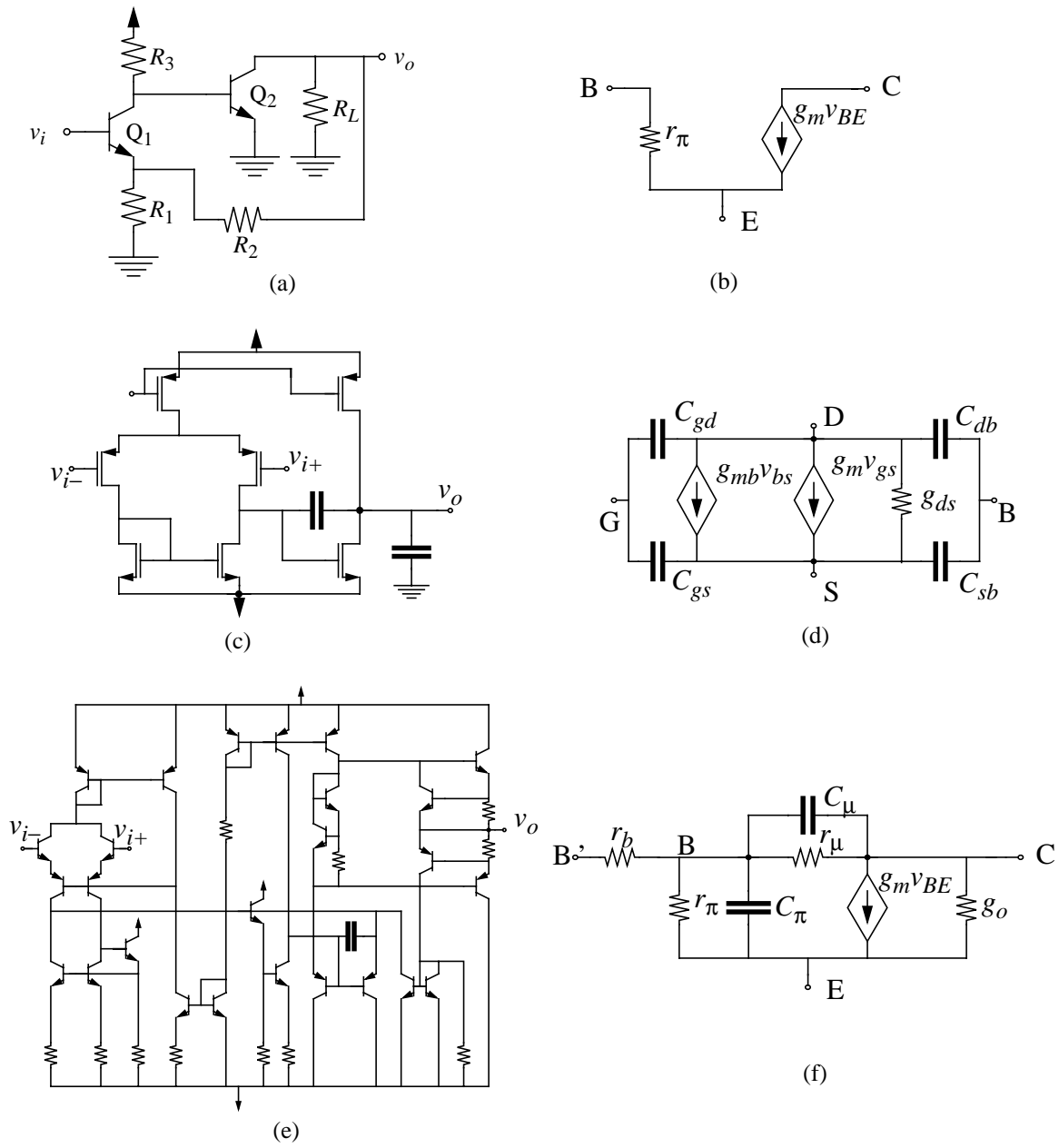


Figure 1 (a) BJT feedback amplifier; (b) low-frequency BJT model; (c) Miller operational amplifier; (d) MOSFET model; (e) $\mu A741$ operational amplifier; (f) BJT model.

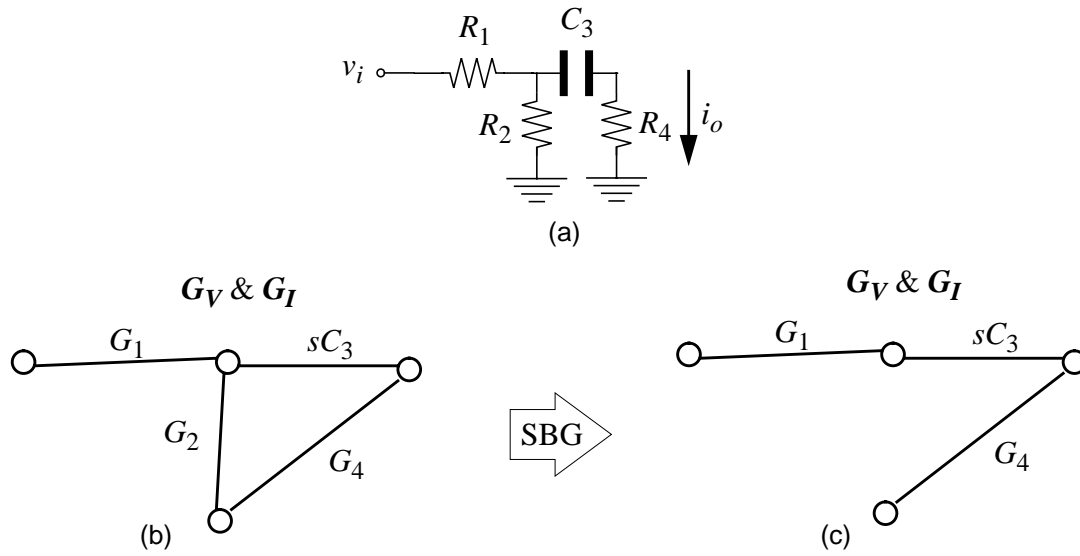


Figure 2 Illustrating an implementation of SBG: (a) Circuit; (b) Formulation of voltage and current graphs; (c) Simplified graphs.

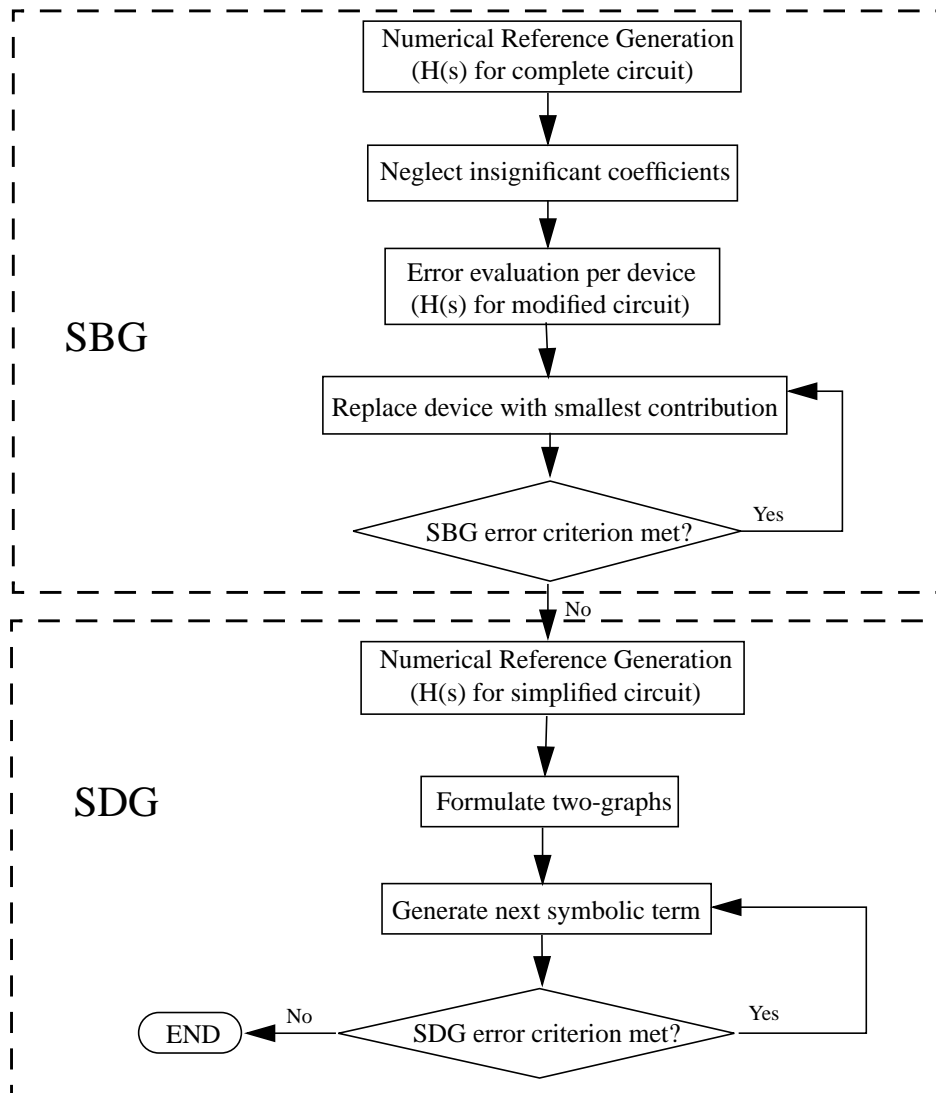


Figure 3 Symbolic analysis methodology for large analog circuits.

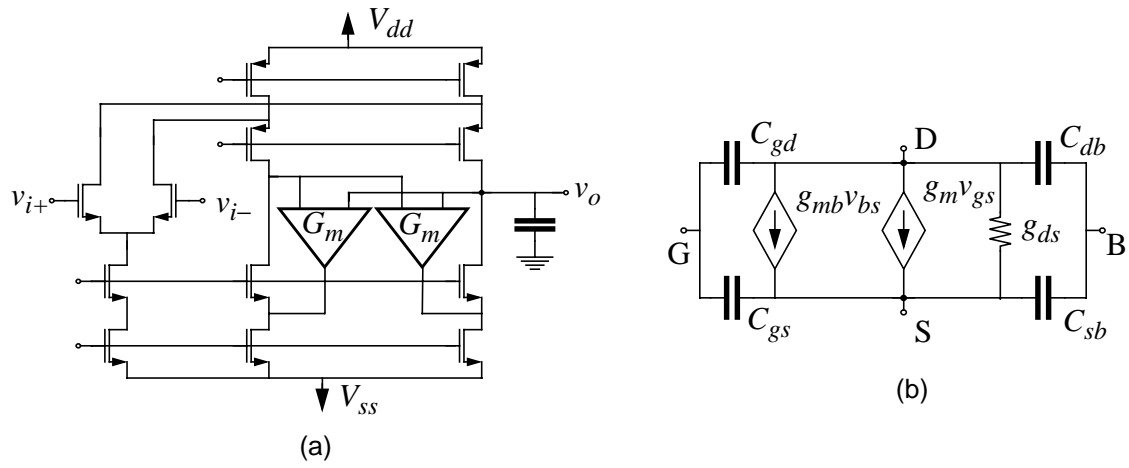


Figure 4 (a) Positive feedback OTA; (b) MOSFET model.

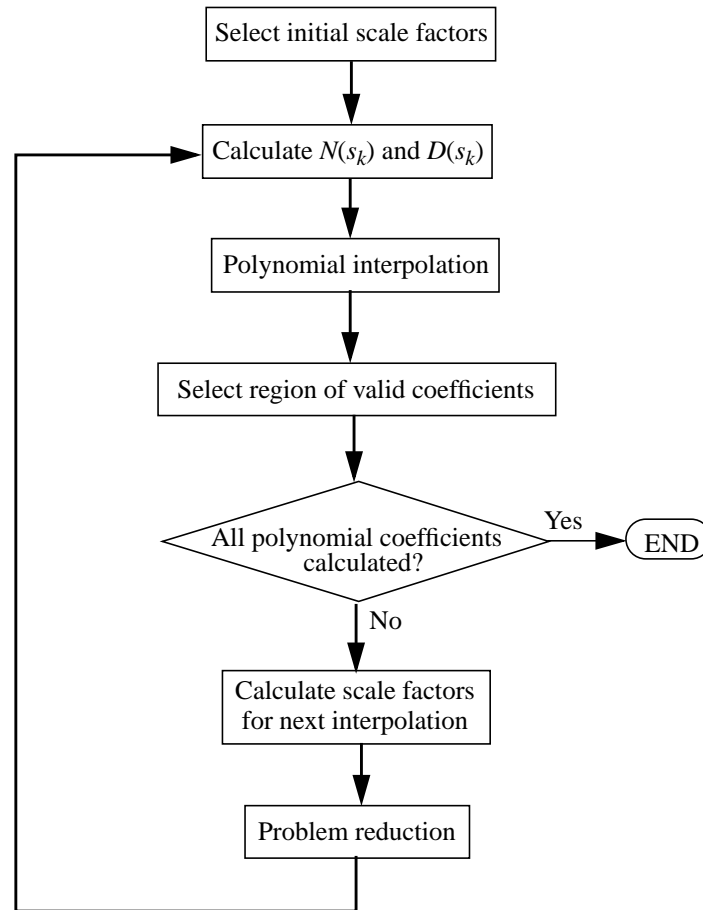


Figure 5 Proposed methodology for numerical reference generation for large analog circuits.

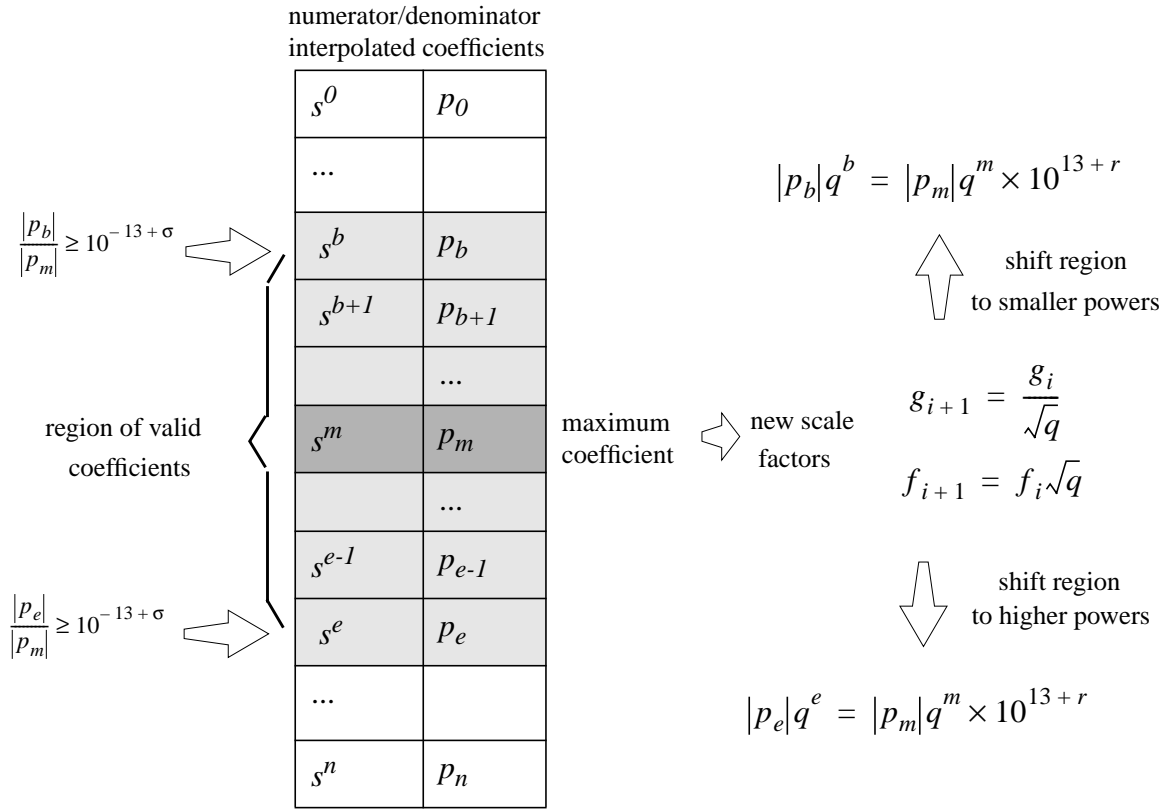


Figure 6 Illustrating the adaptive scaling mechanism at the i -th iteration.

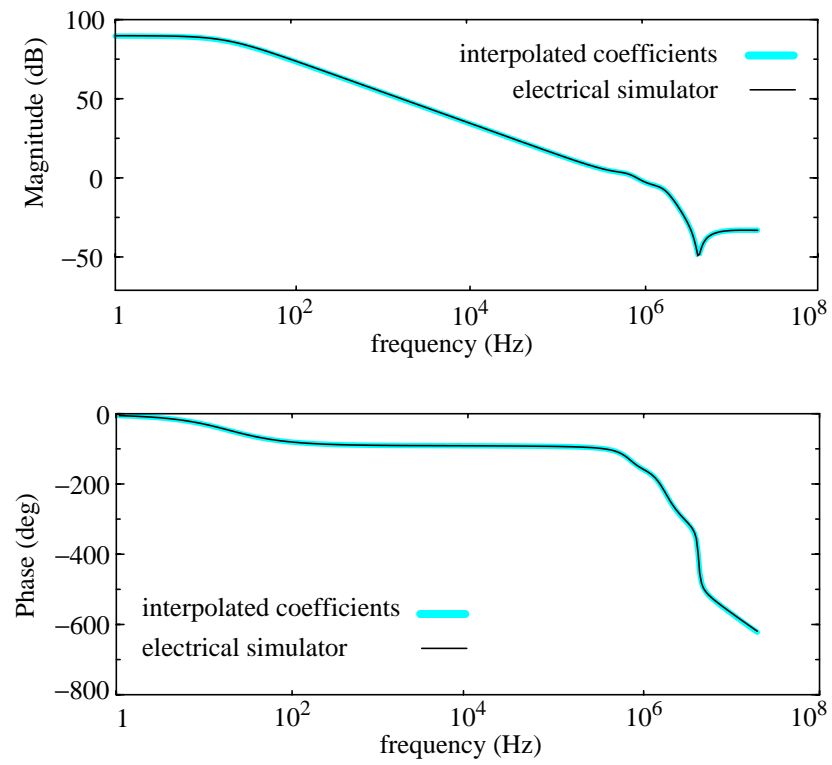


Figure 7 Bode diagrams of the voltage gain of the $\mu A741$ opamp using the interpolated coefficients and an electrical simulator.

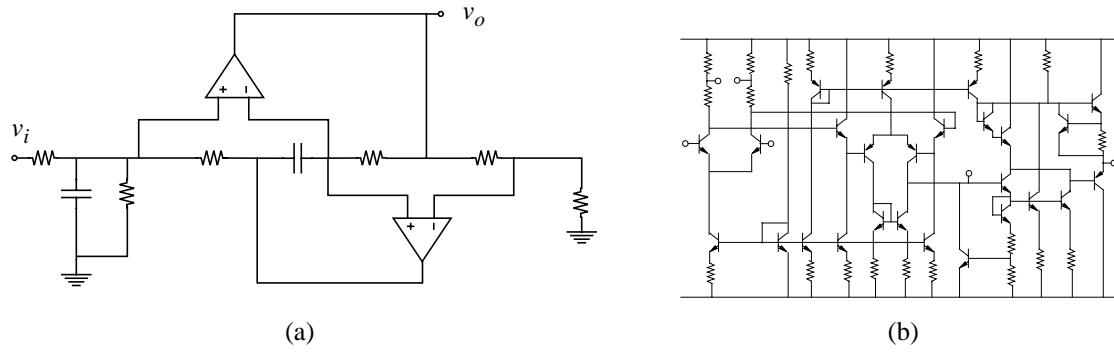


Figure 8 (a) Bandpass biquad; (b) $\mu A725$ opamp.

Table 1 Transfer function coefficients for the differential voltage gain of Fig. 4a using interpolation points on the unit circle.

s^i	Numerator coefficients	Denominator coefficients
s^0	$-5.8296e-25+j0.0$	$+8.9418e-30+j0.0$
s^1	$-1.5484e-33-j2.2958e-41$	$+3.8525e-36-j7.0064e-47$
s^2	$-2.5254e-41+j1.8367e-41$	$+2.3920e-43-j1.4013e-46$
s^3	$-5.5101e-41+j0.0$	$+1.0646e-43-j1.4013e-46$
s^4	$+7.3468e-41+j3.6734e-41$	$-8.4077e-46-j5.6051e-46$
s^5	$-4.5917e-41+j3.5695e-41$	$+2.1019e-45-j5.4751e-46$
s^6	$+5.5101e-41+j4.1326e-41$	$-4.2039e-46-j5.6051e-46$
s^7	$+1.8826e-40-j2.0203e-40$	$+1.0243e-43+j3.0828e-45$
s^8	$-1.1479e-40+j5.5101e-41$	$-1.8020e-43-j5.6051e-46$
s^9	$-1.7448e-40-j1.6530e-40$	$+6.8383e-43+j2.5223e-45$

Table 2 Denominator coefficients for the differential voltage gain of Fig. 4a using a frequency scale factor $f = 10^8$.

s^i	(a) Normalized coefficients	(b) Denormalized coefficients
s^0	8.94186e-30-j5.74242e-44	8.94186e-30-j5.74242e-44
s^1	3.85259e-28+j1.39200e-44	3.85259e-36+j1.39200e-52
s^2	4.20423e-27+j8.17887e-43	4.20423e-43+j8.17887e-59
s^3	1.31933e-27+j3.12804e-43	1.31933e-51+j3.12804e-67
s^4	1.69136e-28-j2.85774e-43	1.69136e-60-j2.85774e-75
s^5	1.09681e-29-j3.09269e-43	1.09681e-69-j3.09269e-83
s^6	3.58708e-31-j4.41609e-43	3.58708e-79-j4.41609e-91
s^7	4.72362e-33+j2.45089e-43	4.72362e-89+j2.45089e-99
s^8	9.71439e-42-j6.61683e-43	9.71439e-106-j6.61683e-107
s^9	-7.50537e-42-j2.6935e-42	-7.50537e-114-j2.6935e-114

Table 3 Valid voltage gain coefficients obtained from the first algorithm iteration on the μ A741 opamp in Fig. 1e.

s^i	Numerator coefficients		Denominator coefficients	
	Normalized	Denormalized	Normalized	Denormalized
s^0	-9.60926e+122	-5.58675e-86	-2.82408e+118	-1.6419e-90
s^1	-1.05987e+124	-2.10393e-91	-7.32222e+122	-1.45352e-92
s^2	-1.48757e+124	-1.00824e-97	-8.26327e+123	-5.60064e-98
s^3	-1.09256e+124	-2.52835e-104	-1.28095e+124	-2.96432e-104
s^4	-4.74222e+123	-3.74701e-111	-1.20867e+124	-9.55018e-111
s^5	-1.20465e+123	-3.24992e-118	-7.46903e+123	-2.015e-117
s^6	-1.7316e+122	-1.59502e-125	-3.17468e+123	-2.92428e-124
s^7	-1.17059e+121	-3.68155e-133	-9.73518e+122	-3.06176e-131
s^8	3.98904e+119	4.28355e-141	-2.19449e+122	-2.3565e-138
s^9	2.12204e+119	7.7803e-148	-3.61682e+121	-1.32608e-145
s^{10}	3.16408e+118	3.96094e-155	-4.2945e+120	-5.37606e-153
s^{11}	2.81205e+117	1.20194e-162	-3.61821e+119	-1.54651e-160
s^{12}	1.45161e+116	...	-2.13624e+118	-3.11759e-168
s^{13}	3.74942e+114	...	-8.7689e+116	...
...	
s^{48}	...			

Table 4 Valid voltage gain coefficients obtained from the second algorithm iteration on the μ A741 opamp in Fig. 1e.

s^i	Numerator coefficient		Denominator coefficient	
	Normalized	Denormalized	Normalized	Denormalized
...	
s^{11}	1.26823e+83	1.20194e-162	-1.6318e+85	-1.54651e-160
s^{12}	2.87085e+84	2.11845e-170	-4.22484e+86	-3.11759e-168
s^{13}	3.25114e+85	1.86795e-178	-7.60487e+87	-4.3694e-176
s^{14}	7.09905e+85	3.17579e-187	-9.50869e+88	-4.25375e-184
s^{15}	-2.41332e+87	-8.40596e-195	-8.31808e+89	-2.89732e-192
s^{16}	-3.10937e+88	-8.4327e-203	-5.16263e+90	-1.40012e-200
s^{17}	-1.93746e+89	-4.09119e-211	-2.31064e+91	-4.8792e-209
s^{18}	-7.5856e+89	-1.24718e-219	-7.57228e+91	-1.24499e-217
s^{19}	-2.03572e+90	-2.60601e-228	-1.84185e+92	-2.35783e-226
s^{20}	-3.91629e+90	-3.90351e-237	-3.36737e+92	-3.35638e-235
s^{21}	-5.55819e+90	-4.31356e-246	-4.68533e+92	-3.63616e-244
s^{22}	-5.94529e+90	-3.5925e-255	-5.02443e+92	-3.03607e-253
s^{23}	-4.87733e+90	-2.29471e-264	-4.20538e+92	-1.97857e-262
s^{24}	-3.11448e+90	-1.14091e-273	-2.78054e+92	-1.01858e-271
s^{25}	-1.56748e+90	-4.47086e-283	-1.46833e+92	-4.18806e-281
s^{26}	-6.28204e+89	-1.39512e-292	-6.25244e+91	-1.38854e-290
s^{27}	-2.02144e+89	-3.49537e-302	-2.1642e+91	-3.74221e-300
s^{28}	-5.2559e+88	-7.0762e-312	-6.12909e+90	-8.25182e-310
s^{29}	-1.10938e+88	-1.16293e-321	-1.4274e+90	-1.49631e-319
s^{30}	-1.9069e+87	-1.55641e-331	-2.7438e+89	-2.23949e-329
s^{31}	-2.67406e+86	-1.69938e-341	-4.36393e+88	-2.7733e-339
s^{32}	-3.06103e+85	-1.51463e-351	-5.74996e+87	-2.84515e-349
s^{33}	-2.85887e+84	-1.10143e-361	-6.27718e+86	-2.41839e-359
s^{34}	-2.17505e+83		-5.67206e+85	-1.70147e-369
s^{35}	-1.34446e+82		-4.2331e+84	
...	

Table 5 Valid voltage gain coefficients obtained from the third algorithm iteration on the $\mu A741$ opamp in Fig. 1e.

s^i	Numerator coefficient		Denominator coefficient	
	Normalized	Denormalized	Normalized	Denormalized
...	
s^{31}	-9.54531e+100	-1.69938e-341	-1.55775e+103	-2.7733e-339
s^{32}	-1.31048e+102	-1.51463e-351	-2.46167e+104	-2.84515e-349
s^{33}	-1.46792e+103	-1.10143e-361	-3.2231e+105	-2.41839e-359
s^{34}	-1.33944e+104	-6.52456e-372	-3.49298e+106	-1.70147e-369
s^{35}	-9.93e+104	-3.14017e-382	-3.12651e+107	-9.88697e-380
s^{36}	-5.95853e+105	-1.22326e-392	-2.30357e+108	-4.72912e-390
s^{37}	-2.87924e+106	-3.83736e-403	-1.39051e+109	-1.85323e-400
s^{38}	-1.11283e+107	-9.62853e-414	-6.83395e+109	-5.91294e-411
s^{39}	-3.40957e+107	-1.91517e-424	-2.7127e+110	-1.52373e-421
s^{40}	-8.18174e+107	-2.98352e-435	-8.6082e+110	-3.13903e-432
s^{41}	-1.51217e+108	-3.57981e-446	-2.15541e+111	-5.10259e-443
s^{42}	-2.10129e+108	-3.2294e-457	-4.18739e+111	-6.43546e-454
s^{43}	-2.11603e+108	-2.11122e-468	-6.17333e+111	-6.1593e-465
s^{44}	-1.45292e+108	-9.41084e-480	-6.70084e+111	-4.34028e-476
s^{45}	-6.06159e+107	-2.54889e-491	-5.12766e+111	-2.15617e-487
s^{46}	-1.15315e+107	-3.14794e-503	-2.58524e+111	-7.05735e-499
s^{47}			-7.60583e+110	-1.34792e-510
s^{48}			-9.74782e+109	-1.1215e-522

Table 6 Valid denominator coefficients of the voltage gain of the bandpass biquad in Fig. 8a obtained from the (a) first, (b) second, (c) third and (d) fourth algorithm iteration.

s^i	(a) Denominator coefficients	
	Normalized	Denormalized
...	...	
s^{23}	-7.51056e+09	
s^{24}	-5.94246e+10	
s^{25}	-7.29364e+11	-5.2904e-395
...
s^{40}	-5.80151e+18	-3.26183e-513
s^{41}	-6.87954e+18	-1.70385e-521
s^{42}	-7.2521e+18	-7.91199e-530
s^{43}	-6.79335e+18	-3.2648e-538
...
s^{58}	-1.47487e+12	-3.23433e-670
s^{59}	-8.03012e+11	-1.96111e-679
...		

s^i	(b) Denominator coefficients	
	Normalized	Denormalized
s^0	-3.91876e+56	
s^1	-1.01152e+58	-4.94769e-230
s^2	-7.75231e+59	-4.44594e-235
...
s^8	-1.04737e+64	-1.56051e-272
s^9	-1.42046e+64	-2.48143e-279
s^{10}	-1.56628e+64	-3.2081e-286
s^{11}	-1.4507e+64	-3.48388e-293
s^{12}	-1.163e+64	-3.2747e-300
...
s^{23}	-2.94228e+60	-4.76909e-380
s^{24}	-8.74623e+59	-1.66218e-387
s^{25}	...calculated...	

s^i	(c) Denominator coefficients	
	Normalized	Denormalized
s^{59}	...calculated...	
s^{60}	-3.64197e+12	-1.05641e-688
...
s^{68}	-1.03089e+15	-1.183e-764
s^{69}	-1.26356e+15	-2.2963e-774
s^{70}	-1.39033e+15	-4.00138e-784
s^{71}	-1.37471e+15	-6.2656e-794
...
s^{86}	-4.32544e+09	-1.94841e-946
s^{87}	-7.99215e+08	-5.70129e-957
s^{88}	-1.41979e+08	-1.49098e-967
s^{89}	-1.94284e+07	
...

s^i	(d) Denominator coefficients	
	Normalized	Denormalized
s^{87}	...calculated...	
s^{88}	...calculated...	
s^{89}	-9.84964e+47	-3.47589e-978
...
s^{94}	-5.04429e+49	-3.76528e-1032
s^{95}	-7.39434e+49	-4.04545e-1043
s^{96}	-9.31988e+49	-3.73721e-1054
s^{97}	-1.00029e+50	-2.9399e-1065
s^{98}	-9.03506e+49	-1.9463e-1076
...
s^{104}	-3.93227e+47	-1.31324e-1145
s^{105}	-4.561e+46	-1.11642e-1157
s^{106}	-2.41567e+45	-4.33388e-1170

LIST OF FIGURES

- Figure 1** (a) BJT feedback amplifier; (b) low-frequency BJT model; (c) Miller operational amplifier; (d) MOSFET model; (e) mA741 operational amplifier; (f) BJT model.
- Figure 2** Illustrating an implementation of SBG: (a) Circuit; (b) Formulation of voltage and current graphs; (c) Simplified graphs.
- Figure 3** Symbolic analysis methodology for large analog circuits.
- Figure 4** (a) Positive feedback OTA; (b) MOSFET model.
- Figure 5** Proposed methodology for numerical reference generation for large analog circuits.
- Figure 6** Illustrating the adaptive scaling mechanism at the i -th iteration.
- Figure 7** Bode diagrams of the voltage gain of the μ A741 opamp using the interpolated coefficients and an electrical simulator.
- Figure 8** (a) Bandpass biquad; (b) μ A725 opamp.

LIST OF TABLES

- Table 1** Transfer function coefficients for the differential voltage gain of Fig. 4a using interpolation points on the unit circle.
- Table 2** Denominator coefficients for the differential voltage gain of Fig. 4a using a frequency scale factor.
- Table 3** Valid voltage gain coefficients obtained from the first algorithm iteration on the μ A741 opamp in Fig. 1e.
- Table 4** Valid voltage gain coefficients obtained from the second algorithm iteration on the μ A741 opamp in Fig. 1e.
- Table 5** Valid voltage gain coefficients obtained from the third algorithm iteration on the μ A741 opamp in Fig. 1e.
- Table 6** Valid denominator coefficients of the voltage gain of the bandpass biquad in Fig. 8a obtained from the (a) first, (b) second, (c) third and (d) fourth algorithm iteration.

FOOTNOTES

1. The number of terms for Fig. 1c was obtained using ASAP [3] while the lower bound of the number of terms for Fig. 1e was calculated using the theory presented in [4].